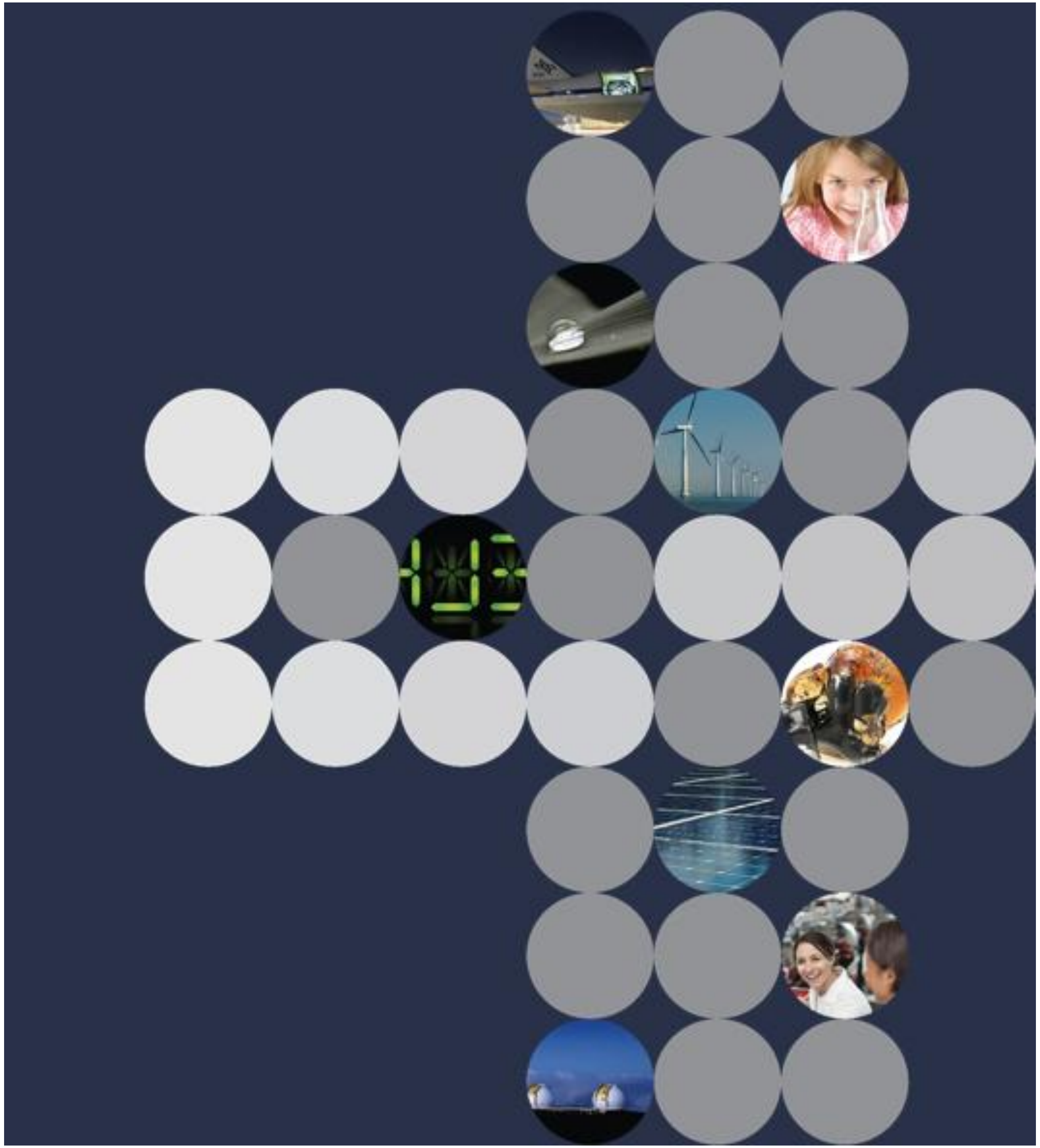
EE512 – Applied Biomedical Signal Processing

Neural network regularization and applications

Clémentine AGUET
CSEM Signal Processing & AI Group

EPFL csem



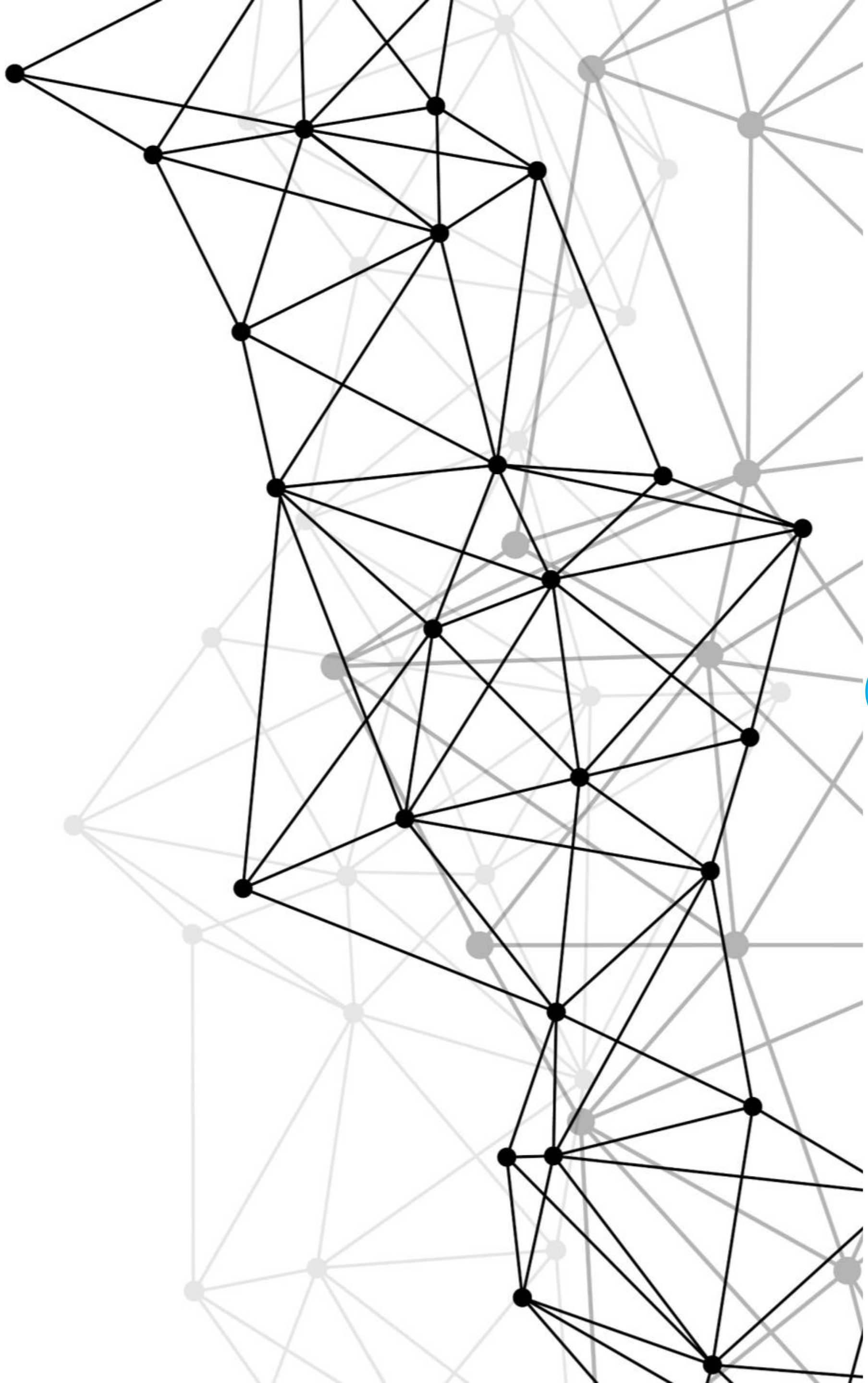
Communication

- Next week
  - No lecture nor lab but Q&A session.
  - Questions should be submitted in advance. Will be announced in Moodle.
  - We will be there to answer your questions on lectures, labs, etc.
- Exam
  - Question on each labs + 4 exercises (similar to mid-term).
  - Open book but no laptop nor smartphone.

EPFL csem

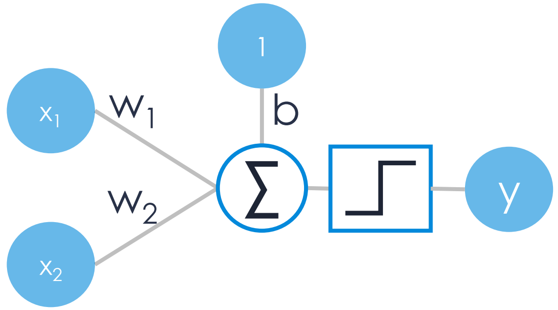
Content

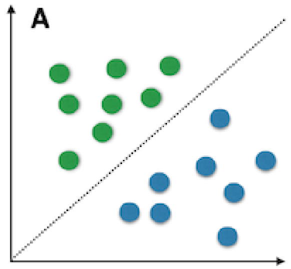
- Neural network recap
- Regularization
- Different architectures
- Labs



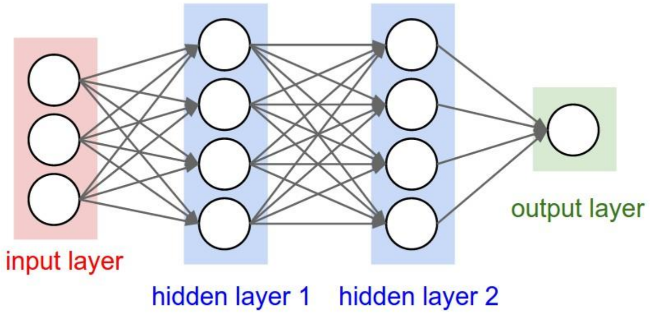
Neural network recap

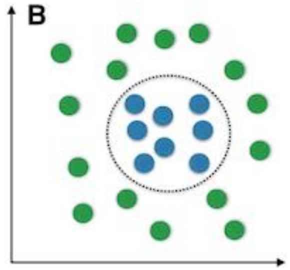
- **Perceptron**
  - Building block of Neural network
  - Linear binary classifier
  - Weighted sum and activation function





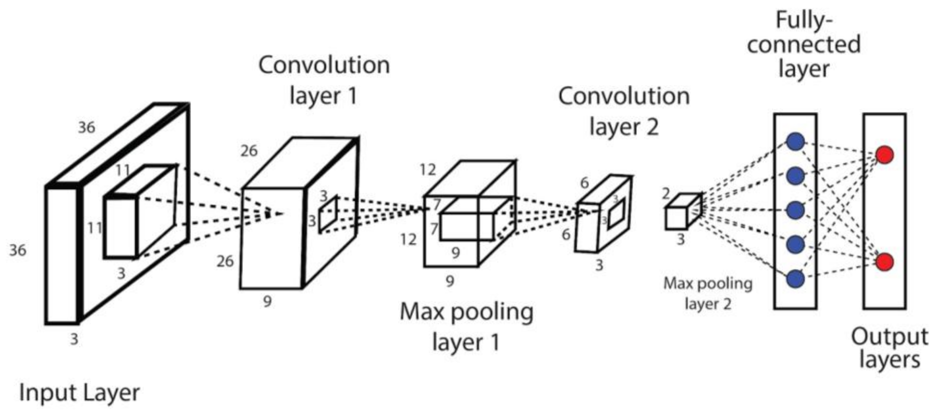
- **Multilayer Perceptron (MLP)**
  - Feedforward NN
  - Solve non-linearly separable problems
  - Series of fully connected layers
  - Each node in a layer is connected to all nodes in next layer
  - Each connection has a weight
  - Deep models = MLP with many layers





Neural network recap

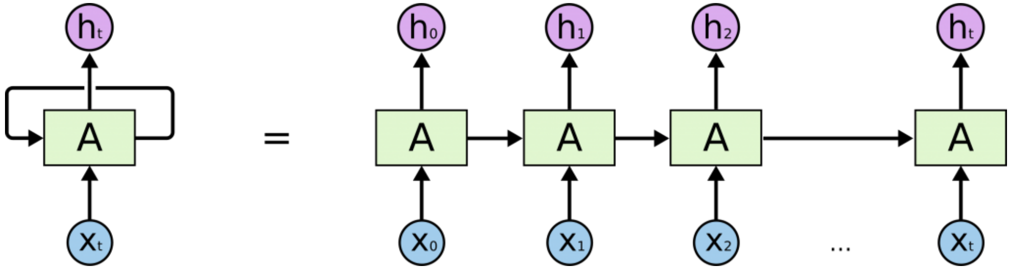
- **Convolutional neural network (CNN)**
  - Computer vision
  - Extract features hierarchically
  - Convolutional layers
    - Extract features
    - Apply set of learnable filters (kernel)
    - Local connectivity and parameter sharing
  - Pooling layers
    - Dimensionality reduction
      - Reduce number of parameters and memory usage
    - Invariance to transformations
  - Fully connected layers
    - Flattened feature maps to 1D
    - Final classification or regression task

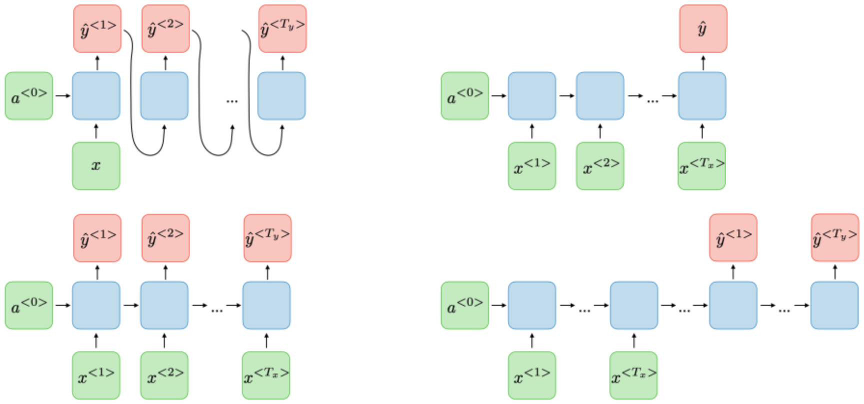


(a) Color (b) Texture (c) Structure

Neural network recap

- **Recurrent neural network (RNN)**
  - Natural language processing, speech recognition
  - Sequential input data
  - Capture temporal representation
  - Weights are shared across time
  - Each neuron has an internal memory
  - Long Short-Term Memory units (LSTM)
  - Gated Recurrent Unit (GRU)





Neural network recap

How neural networks are trained?

- Find the network weights and bias that minimize the empirical loss

$$W^*, b^* = argmin_{W,b} L(W,b)$$

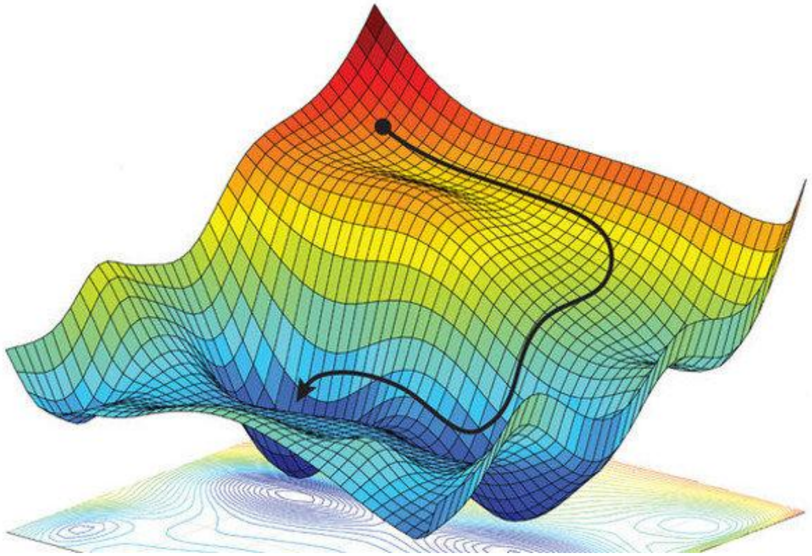
1. Calculate cost function
2. Compute gradient through backpropagation

$$\frac{\partial L(W,b)}{\partial W} \text{ and } \frac{\partial L(W,b)}{\partial b}$$

3. Update parameters

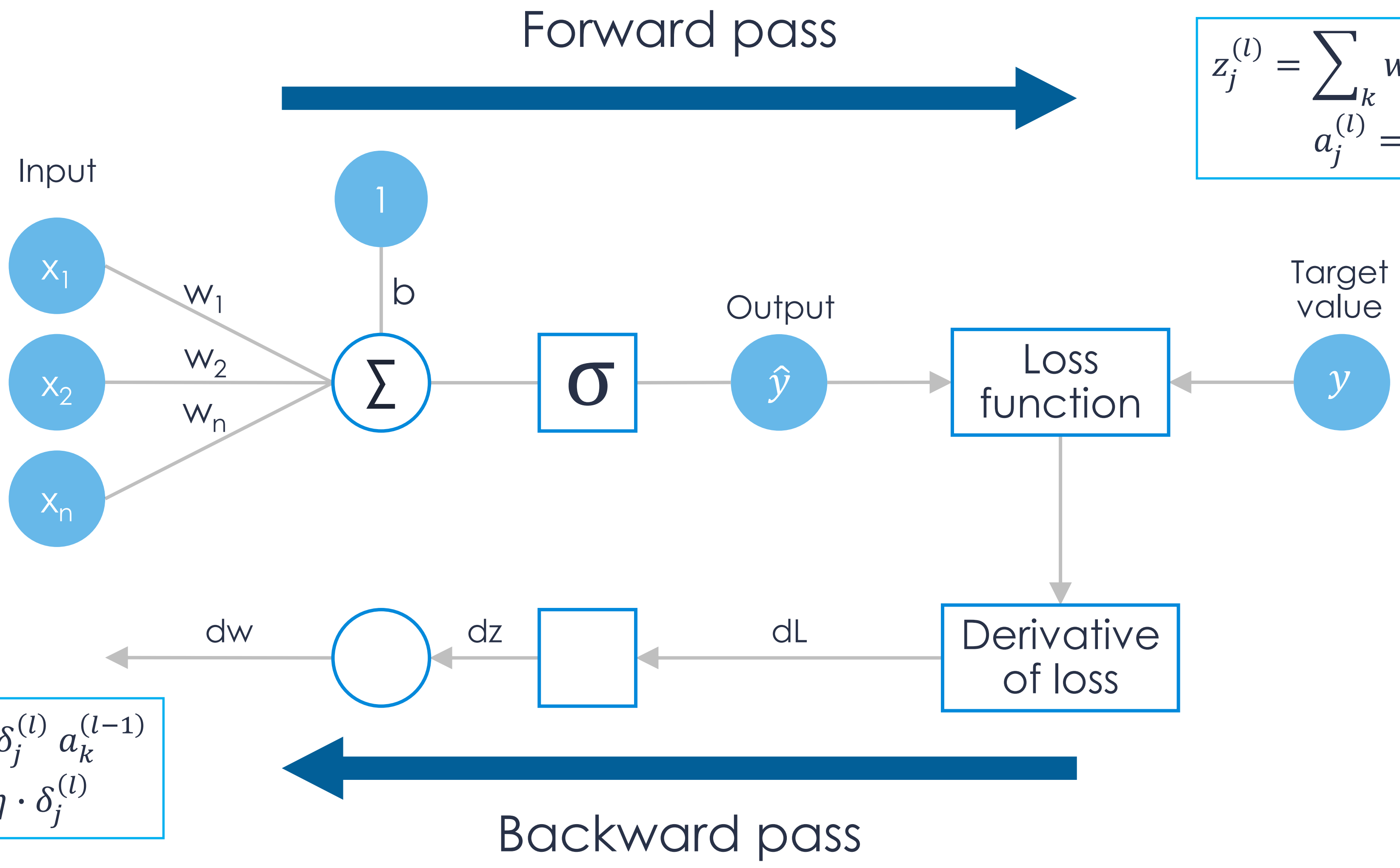
$$W \leftarrow W - \eta \frac{\partial L(W,b)}{\partial W}$$

$$b \leftarrow b - \eta \frac{\partial L(W,b)}{\partial b}$$



Neural network recap – Compute gradient

$$z_j^{(l)} = \sum_k w_{jk}^{(l)} a_k^{(l-1)} + b_j^{(l)}$$
$$a_j^{(l)} = \sigma\left(z_j^{(l)}\right)$$



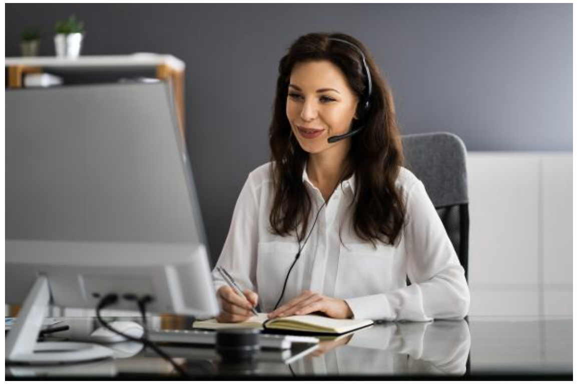
$$w_{jk}^{(l)} \leftarrow w_{jk}^{(l)} - \eta \cdot \delta_j^{(l)}\, a_k^{(l-1)}$$
$$b_j^{(l)} \leftarrow b_j^{(l)} - \eta \cdot \delta_j^{(l)}$$

Neural network recap

- Applications

Speech recognition

Autonomous driving cars



Natural language processing

Stock market prediction

Fraud detection

Healthcare

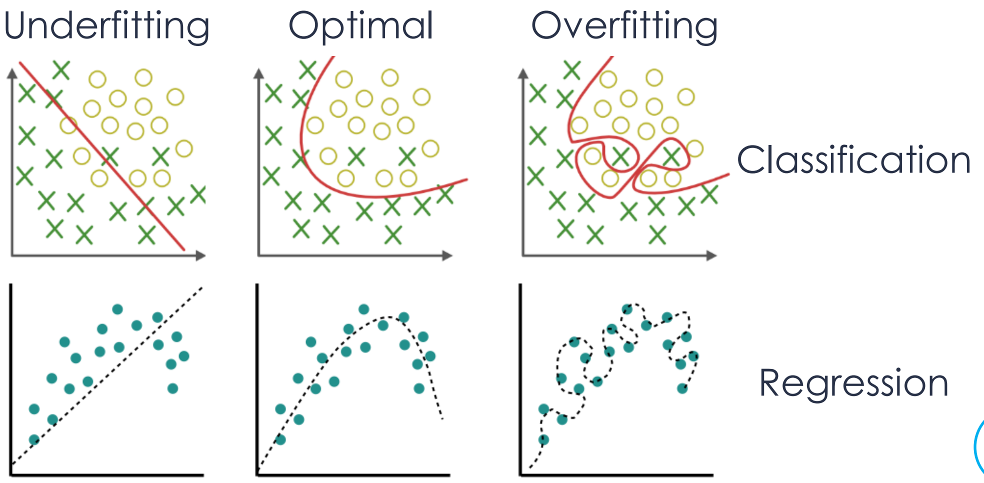
Content

- Neural network recap
- Regularization
- Different architectures
- Labs

Regularization

Why?

- Neural networks are prone to **overfitting**
- Model does not generalize well on unseen data
- Low training error but high validation error

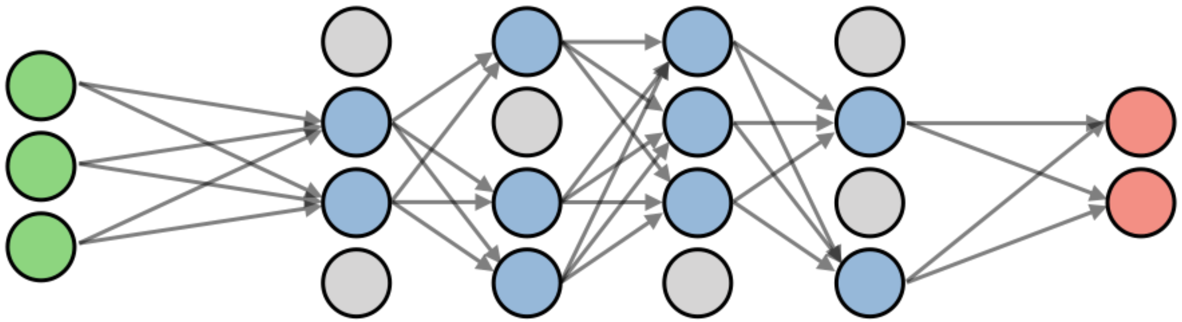


- **Regularization**
    - Slight modifications to learning algorithm such that the model generalizes better
    - Lower the complexity
    - Combine multiple techniques

EPFL csem

Regularization – Dropout

- Avoid relying too much on particular nodes → Learn robust and generalized features
- At every **training** iteration, randomly remove some nodes
- Each iteration has a different set of nodes
- Each node has a probability $p > 0$ to be turned off
- p is an hyperparameter (typically between 0.2 and 0.5)
- Full capacity during inference → All nodes used to make predictions
- More efficient in fully connected layers than convolutional layers

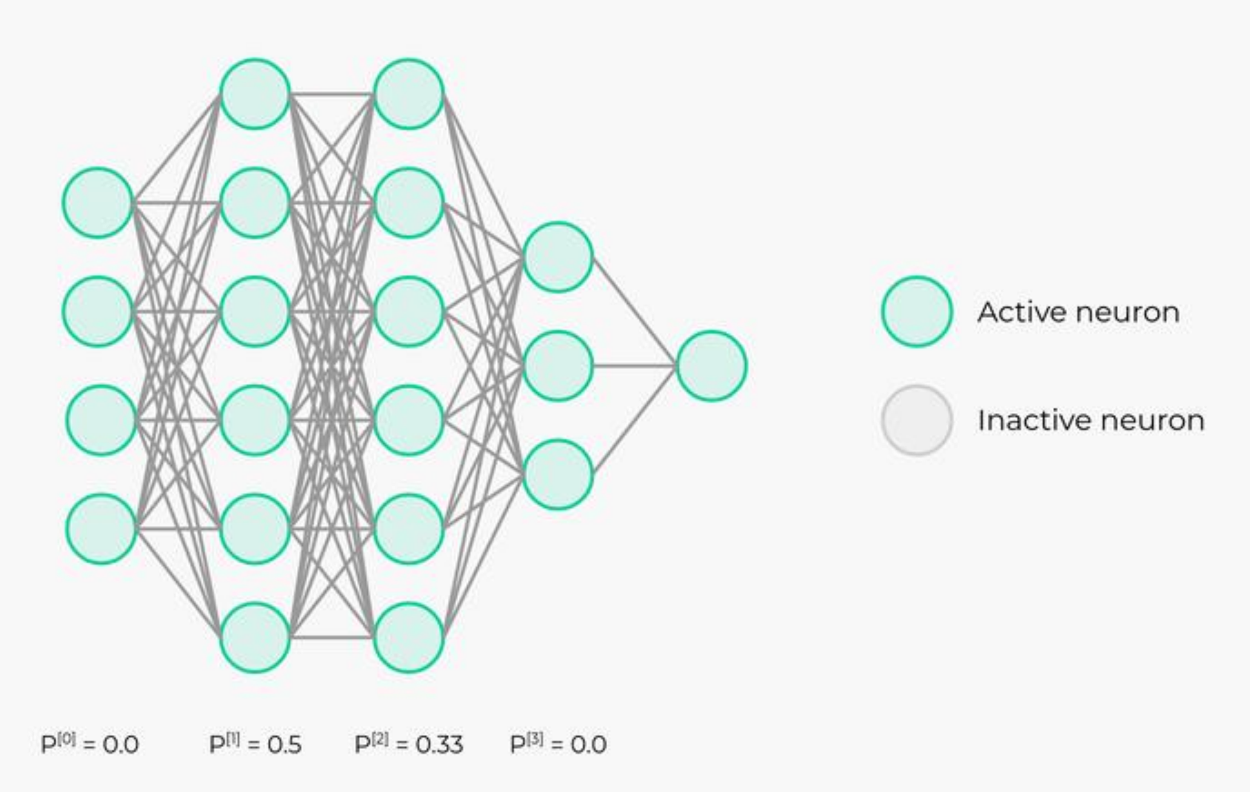


- PyTorch: `torch.nn.Dropout(p=0.5)`

Regularization – Dropout

+ Efficient in large networks
+ Easy to implement
+ Computationally efficient
+ Robustness in feature learning

– Stochasticity → Slower convergence
– Not as effective in small networks
– Hyperparameter to tune



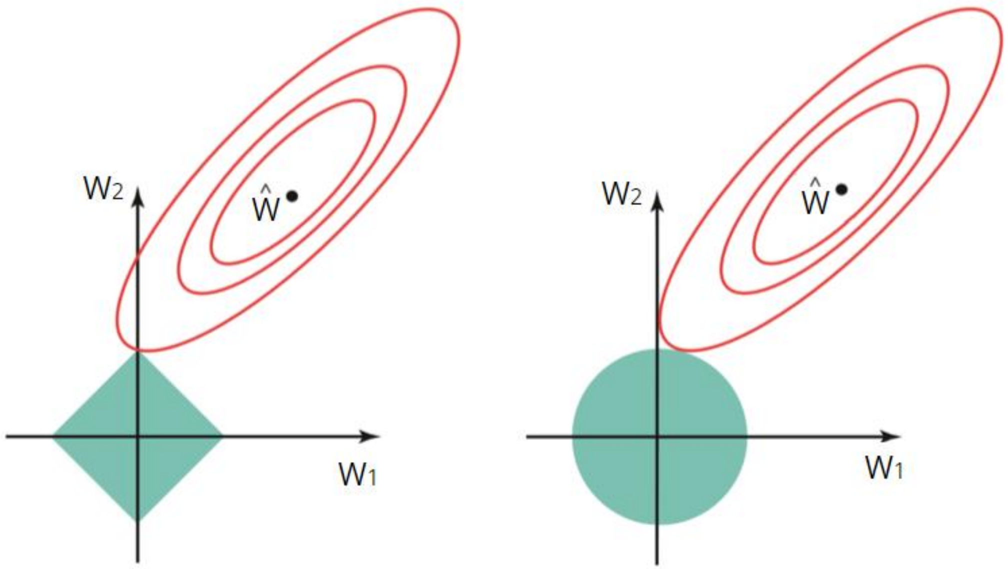
Regularization – L1 & L2 regularization

- Add regularization term to the cost function
- Used as penalty

$$Loss = Loss + Regularization\ term$$

- Ensure that the weights are not too large
- Smaller weights lead to simpler models → Avoid overfitting

L1 regularization



L2 regularization

Regularization – L1 & L2 regularization

- **L2 regularization**
  - Weight decay or Ridge regression
  - Regularization term $\Omega(W)$ is defined as Euclidean norm or L2 norm

$$\Omega(W) = \|W\|_2^2 = \sum_i \sum_j w_{ij}^2$$

$$\hat{L}(W,b) = L(W,b) + \frac{\lambda}{2}\|W\|_2^2$$

  - Regularization term is weighted by a scalar λ and divided by 2
  - λ is the regularization rate
  - Compute gradient

$$\frac{\partial \hat{L}(W,b)}{\partial W} = \frac{\partial L(W,b)}{\partial W} + \lambda W$$

  - Weights update

$$W \leftarrow W - \eta \frac{\partial \hat{L}(W,b)}{\partial W} = (1 - \eta\lambda)W - \eta \frac{\partial L(W,b)}{\partial W}$$

Regularization – L1 & L2 regularization

- **L2 regularization**
  + General regularization in most NN setups
  + Promoting smoothness → Penalizes large weights and generalizable model
  + Differentiable penalty
  + Computational efficiency
  + Handling multicollinearity
  + Unique solution

  – Hyperparameter to tune (regularization rate)
  – Lack of sparsity → No feature selection
  – Sensitivity to irrelevant features

Regularization – L1 & L2 regularization

- **L1 regularization**
  - Lasso regression
  - Regularization term $\Omega(W)$ is defined as L1 norm (Manhattan distance)

$$\Omega(W) = \|W\|_1 = \sum_i \sum_j |w_{ij}|$$

$$\hat{L}(W,b) = L(W,b) + \lambda\|W\|_1$$

  - Regularization term is weighted by a scalar λ
  - λ is the regularization rate
  - Compute gradient

$$\frac{\partial \hat{L}(W,b)}{\partial W} = \frac{\partial L(W,b)}{\partial W} + \lambda sign(W)$$

  - Weights update

$$W \leftarrow W - \eta \frac{\partial \hat{L}(W,b)}{\partial W} = W - \eta\lambda sign(W) - \eta \frac{\partial L(W,b)}{\partial W}$$

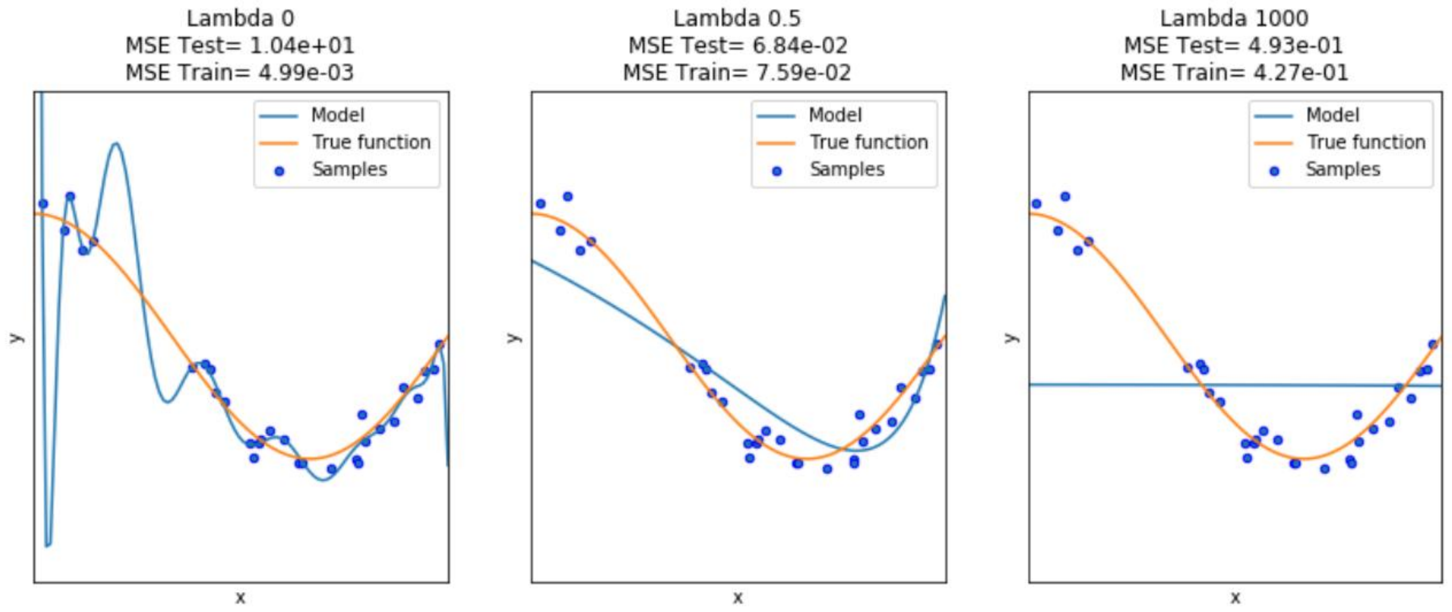
Regularization – L1 & L2 regularization

- **L1 regularization**
  - \+ Feature selection through sparsity
  - \+ Remove irrelevant features → Simpler model
  - \+ Enhance interpretability
  - \+ Robust to outliers
  - \+ Handling high-dimensional data
  - \+ Computational efficiency
  - \+ Simple and widely applicable

  - – Hyperparameter to tune (regularization rate)
  - – Unstable feature selection
  - – Correlated features
  - – Non-differentiability → Complicate optimization (sub-gradient methods)

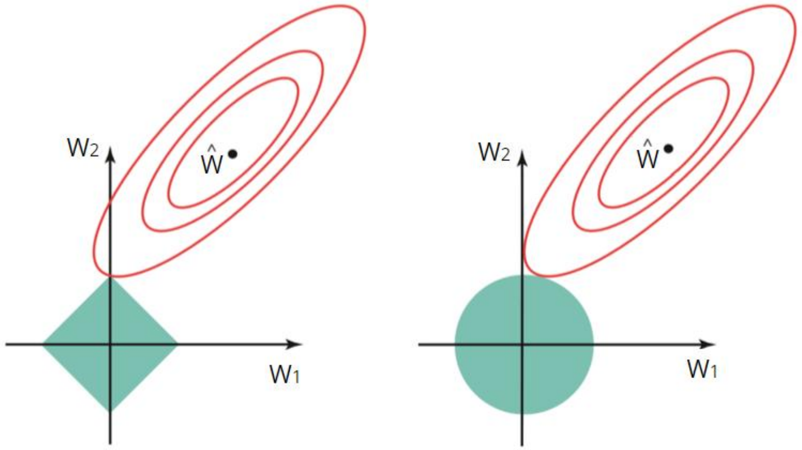
Regularization – L1 & L2 regularization

- **Regularization rate (λ)**
  - Hyperparameter to tune
  - Should be chosen carefully
    - Too high value → Model simpler but increased risk of underfitting
    - Too small value → Model more complex and increased risk of overfitting



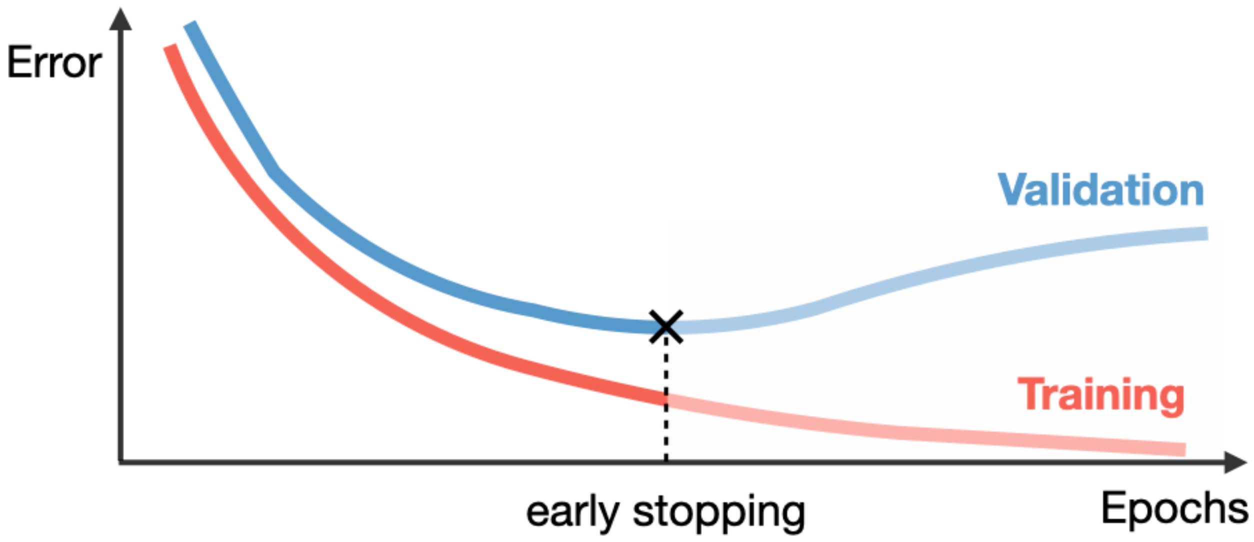
Regularization – L1 & L2 regularization

L1 regularization	L2 regularization
Sum of absolute value of weights	Sum of square of weights
Sparse solution	Non-sparse solution
Built in feature solution	No feature selection
Multiple solutions	One solution
Robust to outliers	Not robust to outliers



Regularization – Early stopping

- Training for too many epochs can lead to overfitting
- Stop training at maximum generalization
- Make sure the model does not learn the noise in the training data
- Kind of cross-validation strategy
- After each epoch, compute error on unseen data (validation data)
- Stop training as soon as the validation loss reached a plateau or starts to increase



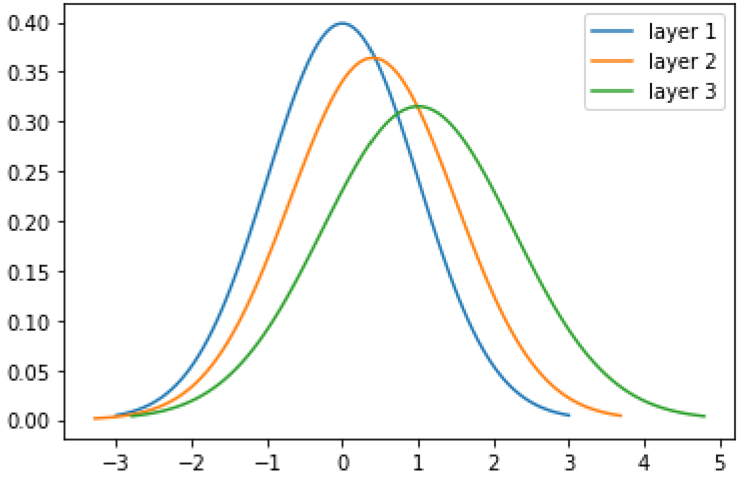
Regularization – Early stopping

+ Computational efficiency
+ No architecture modification
+ Simplicity

– Dependency on validation set
– Hyperparameter to tune (patience)
– Potential underfitting

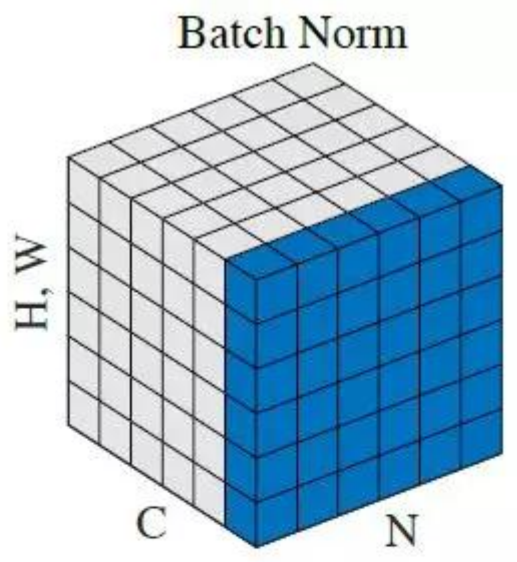
Regularization – Batch normalization

- Internal covariate shift
  - Shift in input distribution over layers
  - Slow learning
  - Deeper network → Amplified effect



Regularization – Batch normalization

- Reduce internal covariate shift
- Smooth the loss landscape
- Improve speed, performance and stability of NN
- Normalize inputs of each layer
    - Mean activation output zero and unit variance
    - Run over batch axis (N)



- After a fully connected or convolutional layer and before non-linearity layer (activation function)
- PyTorch: `torch.nn.BatchNorm1d(num_features, eps=1e-05)`

Regularization – Batch normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\dots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

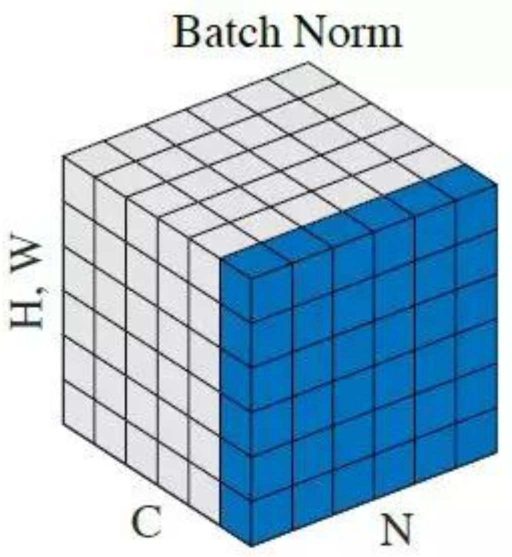
$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Loffe and Szegedy (2015), Batch normalization: accelerating deep network training by reducing internal covariate shift



EPFL csem

Regularization – Batch normalization

+ Accelerate training → Faster convergence
+ Higher learning rate
+ Improved gradient flow → Reduced vanishing/exploding gradients
+ Reduce sensitivity to initialization

– Dependency on mini-batch size
– Increase computational overhead
– Inconsistent behavior between training and inference
– Complexity in recurrent networks

Regularization – Data augmentation

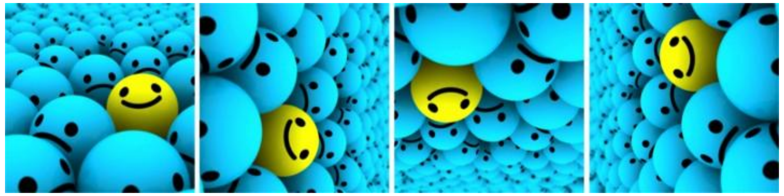
- NNs need a lot of data to be properly trained
- More information can be extracted from original data through augmentation
- Increase amount of data
  - Synthetic data → Generated artificially (GANs for example)
  - Augmented data → Derived from original images

- **Data augmentation**
  - Mostly used with images
  - Get more data from existing ones
  - Make minor changes

Regularization – Data augmentation

- Data augmentation for **images**

Flip



Rotation

Crop

Add noise

- Others
  - Color shift
  - Information loss

Regularization – Data augmentation

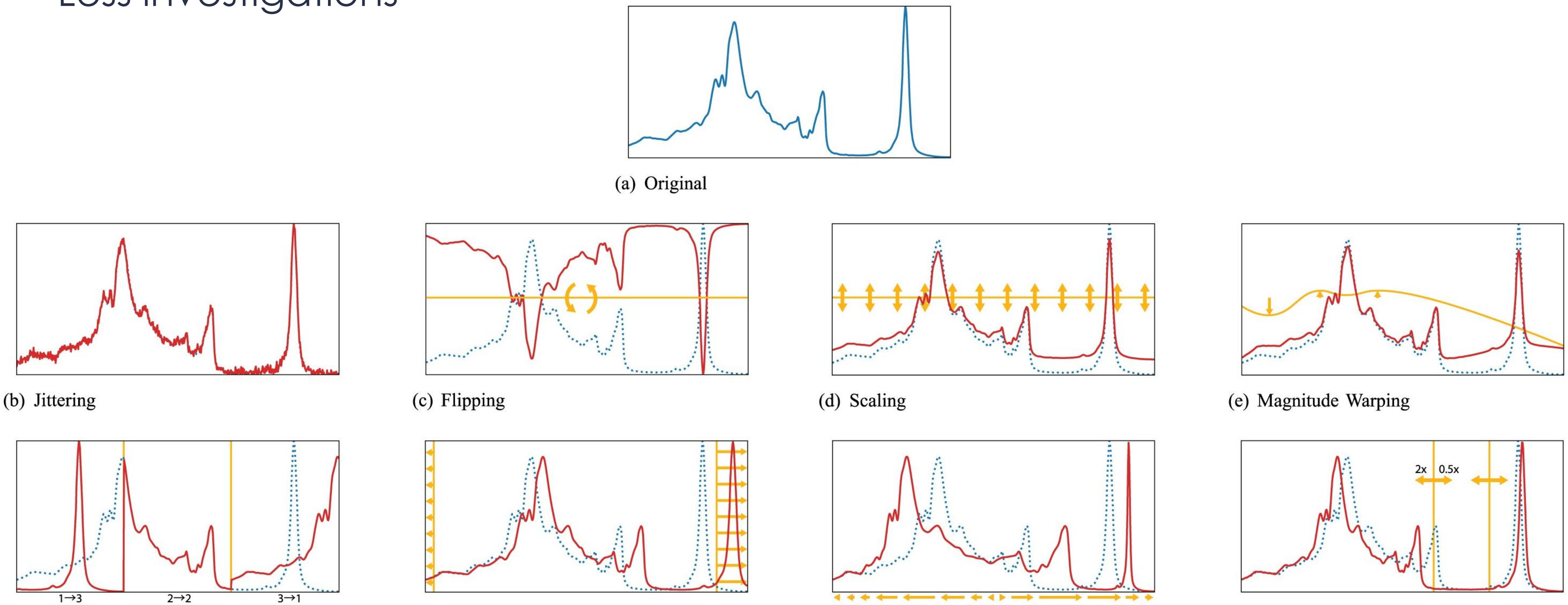
PyTorch function:

```
import torch
from torchvision import transform

transform = transforms.Compose([
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.RandomVerticalFlip(p=0.5),
    transforms.RandomRotation(degrees, interpolation),
    transforms.Resize(size, interpolation),
    tranforms.RandomCrop(size, padding),
    transforms.ToTensor()
    ])
```

Regularization – Data augmentation

- Data augmentation for **time series**
  - Some geometric transformations might change the signal properties
  - Less investigations



(a) Original

(b) Jittering (c) Flipping (d) Scaling (e) Magnitude Warping

(f) Permutation (g) Window Slicing (h) Time Warping (i) Window Warping

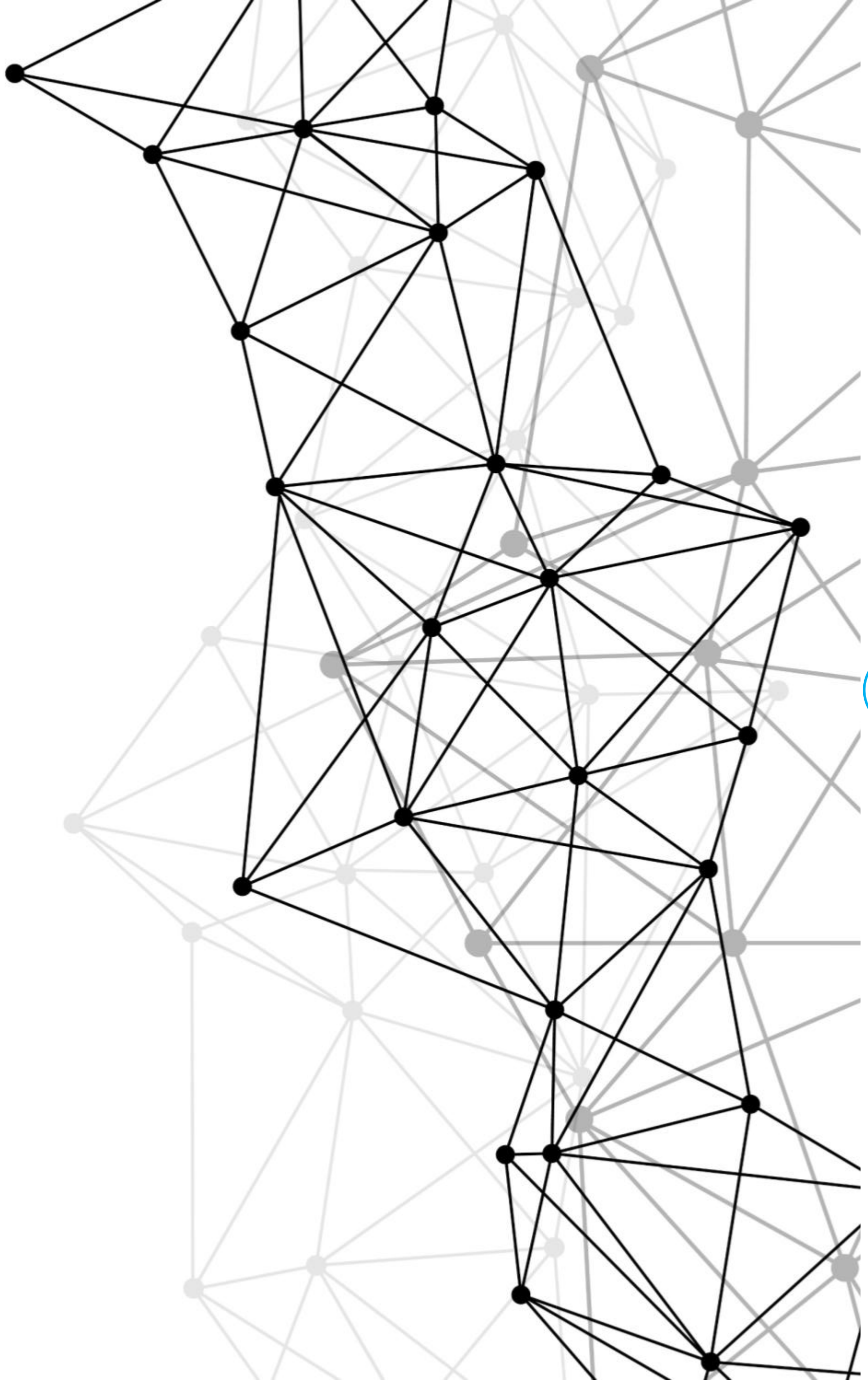
Regularization – Data augmentation

+ Introduction of invariance and robustness
+ Reduced need for costly data collection
+ No modification to the model architecture

– Potential introduction of noise and irrelevant variations
– Risk of label inconsistency
– Domain-specific design and expertise required
– Increased computational overhead
– Not universally applicable across all data types

Content

- Neural network recap
- Regularization
- Different architectures
- Labs



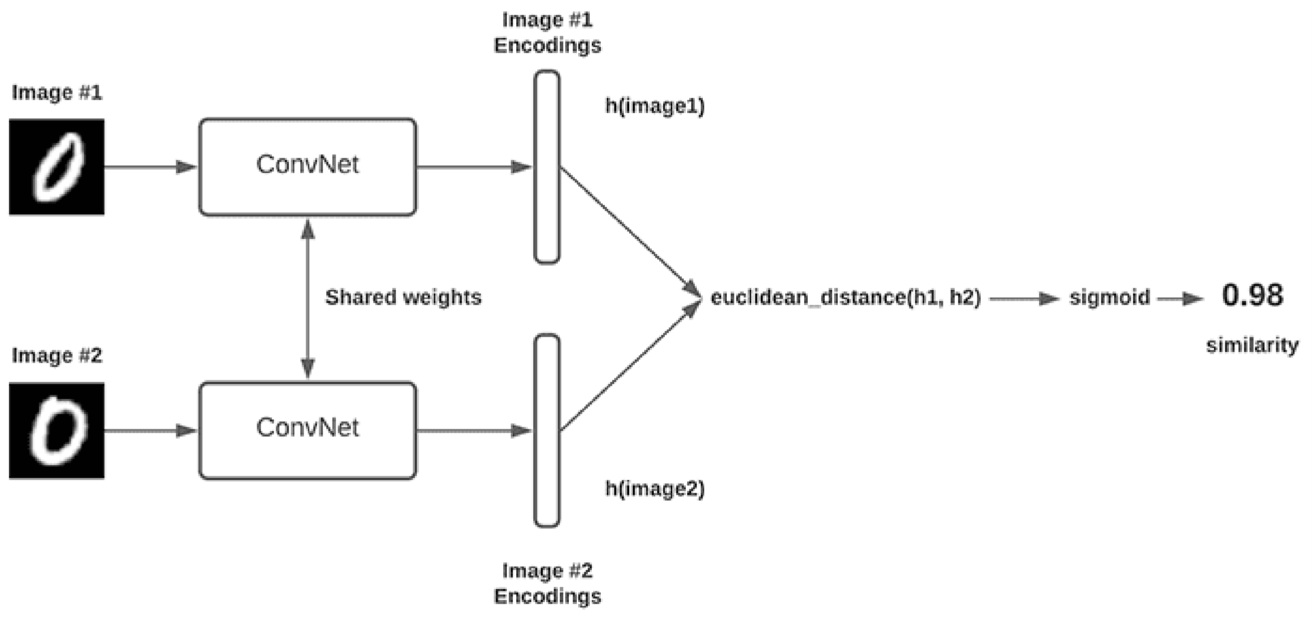
Content

- Different architectures
  - Siamese neural network
  - Generative adversarial network
  - Explainable neural network

Siamese neural network

- Two or more identical subnetworks sharing the same parameters and weights
- Learn representations to compare inputs similarity
- Learn embedding space where distances correspond to semantic similarity
  1. Inputs are passed into twin networks simultaneously
  2. Each subnetwork extracts features from its respective input
  3. Features are combined to measure distance/similarity between inputs

+ Learning from few examples
+ More robust to class imbalance
+ Weight sharing
– Training complexity
– Distance function sensitivity



EPFL csem

Siamese network – Loss function

- **Contrastive loss**
  - Binary similar/dissimilar constraint

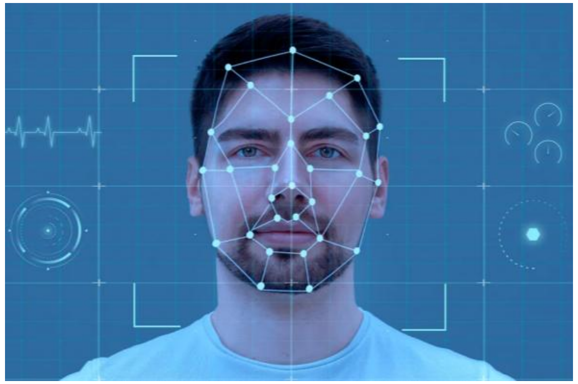
$$L(A,B) = \underbrace{y\|f(A) - f(B)\|_2}_{\substack{\text{Positive pair} \\ d(A,B) \text{ is small}}} + (1-y)\underbrace{max(0, m^2 - \|f(A) - f(B)\|_2)}_{\substack{\text{Negative pair} \\ d(A,B) \text{ is large} \\ \text{Hinge loss}}}$$

- **Triplet loss**
  - Anchor, positive (same class as anchor), negative (different class as anchor)
  - A must be closer to P than to N by as least margin m

$$L(A,P,N) = max(0, \|f(A) - f(P)\|_2 - \|f(A) - f(N)\|_2 + m)$$

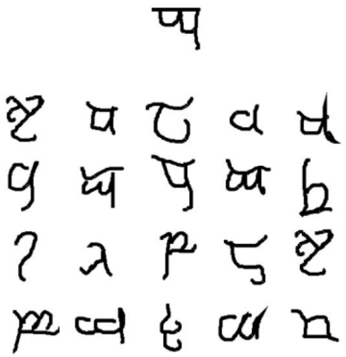
EPFL csem

Siamese network - Applications

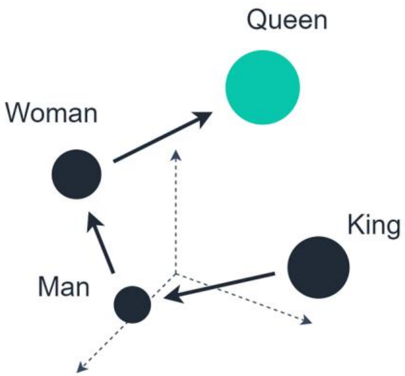


Face recognition

Signature verification



One-shot learning



Text similarity

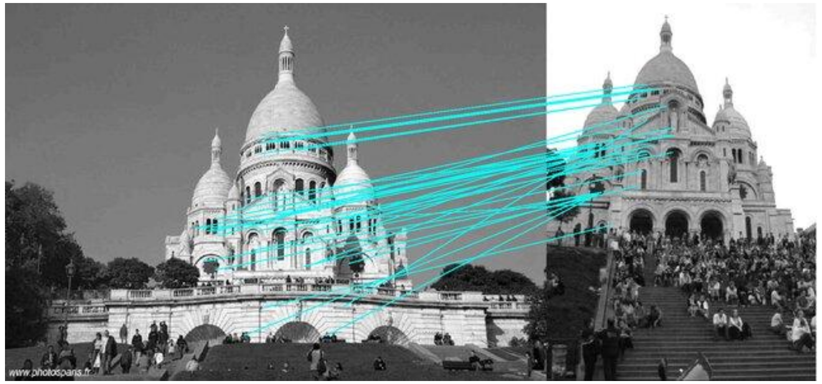


Image matching

Siamese network

Schlesinger et al. 2020, Blood pressure estimation from PPG signals using convolutional neural networks and Siamese network

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

- GOAL
  - Estimate blood pressure (BP) from photoplethysmography (PPG) signal
  - Feature learning approach
    → Automatic feature extraction
  - Siamese architecture

BLOOD PRESSURE ESTIMATION FROM PPG SIGNALS USING CONVOLUTIONAL NEURAL NETWORKS AND SIAMESE NETWORK

Oded Schlesinger[*1], *Nitai Vigderhouse*[*1], *Danny Eytan*[†], *Yair Moshe*[*]

[*]Signal and Image Processing Laboratory (SIPL)
Andrew and Erna Viterbi Faculty of Electrical Engineering
Technion – Israel Institute of Technology
http://sipl.technion.ac.il/

[†]Ruth and Bruce Rappaport Faculty of Medicine
Technion – Israel Institute of Technology

ABSTRACT

Blood pressure (BP) is a vital sign of the human body and an important parameter for early detection of cardiovascular diseases. It is usually measured using cuff-based devices or monitored invasively in critically-ill patients. This paper presents two techniques that enable continuous and noninvasive cuff-less BP estimation using photoplethysmography (PPG) signals with Convolutional Neural Networks (CNNs). The first technique is calibration-free. The second technique achieves a more accurate measurement by estimating BP changes with respect to a patient's PPG and ground truth BP values at calibration time. For this purpose, it uses Siamese network architecture. When trained and tested on the MIMIC-II database, it achieves mean absolute difference in the systolic and diastolic BP of 5.95 mmHg and 3.41 mmHg respectively. These results almost comply with the AAMI recommendation and are as accurate as the values estimated by many home BP measuring devices.

Index Terms— Blood pressure, convolutional neural network (CNN), noninvasive, photoplethysmography (PPG), Siamese network

1. INTRODUCTION

Blood pressure (BP) is the result of force exerted in the arteries by blood as it circulates. It is usually expressed in terms of systolic pressure (when the heart beats and BP is at its highest) and diastolic pressure (between heart beats, when BP is at its lowest) and measured in millimeters of mercury (mmHg). Normal resting BP in an adult is approximately 120 mmHg systolic and 80 mmHg diastolic. BP is an important parameter of the human body whose measurement allows for the early detection of medical issues, especially cardiovascular diseases, which are a leading cause of mortality and morbidity worldwide. High BP, hypertension, is a major risk for dangerous health conditions such as stroke or heart attack. Low BP, hypotension, can cause dizziness and fainting or may indicate serious heart, endocrine or neurological disorders. Therefore, it is highly important to measure BP routinely. Continuous monitoring of BP, along with monitoring of other vital signs, allows an accurate evaluation of the patient's physiological state, prompt detection of deteriorations and their prediction. The current widespread BP monitoring methods are divided into invasive and noninvasive methods. Invasive arterial line is a clinical standard for continuous high accuracy BP measurement. However, it has adverse effects associated with invasive measurements, such as potential infection, all of them are associated with an increased morbidity. Noninvasive BP measurement methods typically use an oscillometry inflatable arm or wrist cuff. These methods are not feasible for long-term ambulatory BP monitoring due to discomfort caused by repeated inflation and deflation and mobility limitations caused by the measuring device.

The Association for the Advancement of Medical Instrumentation (AAMI) recommends that the mean absolute difference (MAD) of noninvasive BP measurement technologies should not be greater than 5 mmHg and the standard deviation should not be greater than 8 mmHg compared to a reference method [1]. Home BP measuring devices may be inaccurate in 5% to 15% of patients, and a difference of 5 mmHg or higher is common [2, 3]. Therefore, BP measurements of other noninvasive technologies are expected to have at least similar accuracy, so they are at least as reliable as home BP measuring devices.

Photoplethysmography (PPG) is an optically obtained signal that can be used to detect blood volume changes in the microvascular bed of tissue. It is obtained by illuminating the skin and measuring changes in light absorption. In a clinical setting, this signal is often obtained by a pulse oximeter while in an ambulatory setting, this signal can be obtained by a smartwatch or other mobile devices. PPG signals contain information on cardiovascular parameters such as heart rate, blood oxygen saturation and BP. Since PPG is noninvasive, simple and low-cost, it has wide potential for clinical

Siamese network – Clinical relevance

- **Blood pressure BP**
- #1 risk factor of cardiovascular diseases
- 1.28 billon people affected
- 46% no noticeable symptoms


Occlusive
Intermittent
Uncomfortable

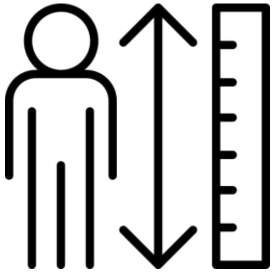
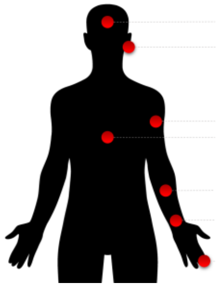
Cuff-based BP measurement

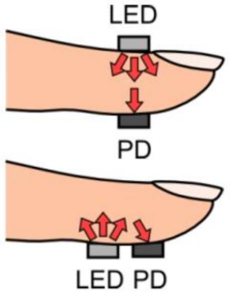
Photoplethysmography PPG

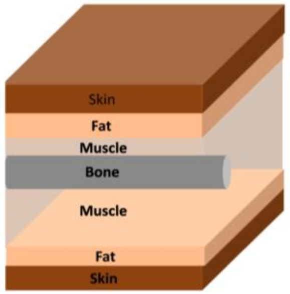

- Measure of changes in light absorption
- Related to blood volume variations
- Light source to illuminate tissue
- Photodetector to measure changes in light intensity
- Information on cardiovascular system
  - Blood oxygen saturation
  - Heart rate
  - Blood pressure

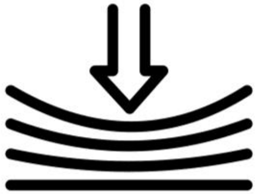
Siamese network - Context

- PPG waveform variability due to individual-specific characteristics or external factors







- Global model often fails to generalize well → Need for per-person calibration
  - Measure taken at the doctor office
  - Partly or completely retraining the model

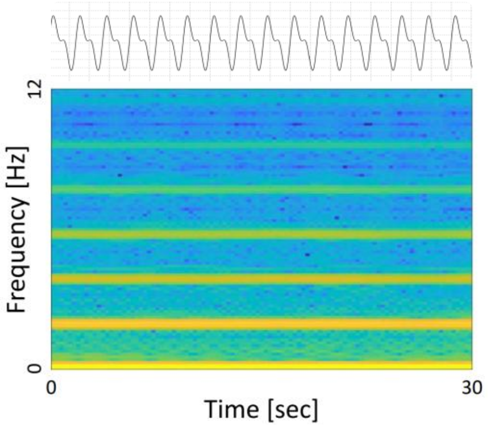
Siamese network – Data

- **MIMIC-II database** [1]
  - 1459 patients in intensive care unit (ICU)

PPG

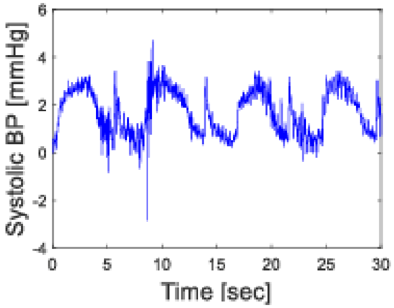
BP arterial line

spectrogram
30-second
PPG window

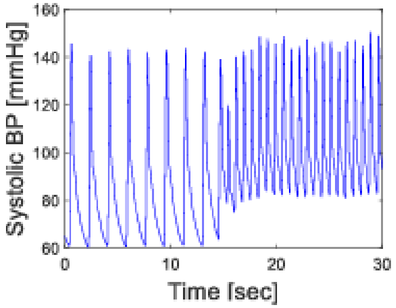


- **Preprocessing**
  - Remove unreliable windows
  - Remove unreliable patients
  - Remove outliers

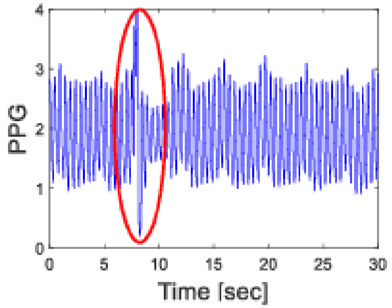
$\sim 10^5$ 30s windows
from 304 patients



No physiological BP



BP fluctuation
within 30s window



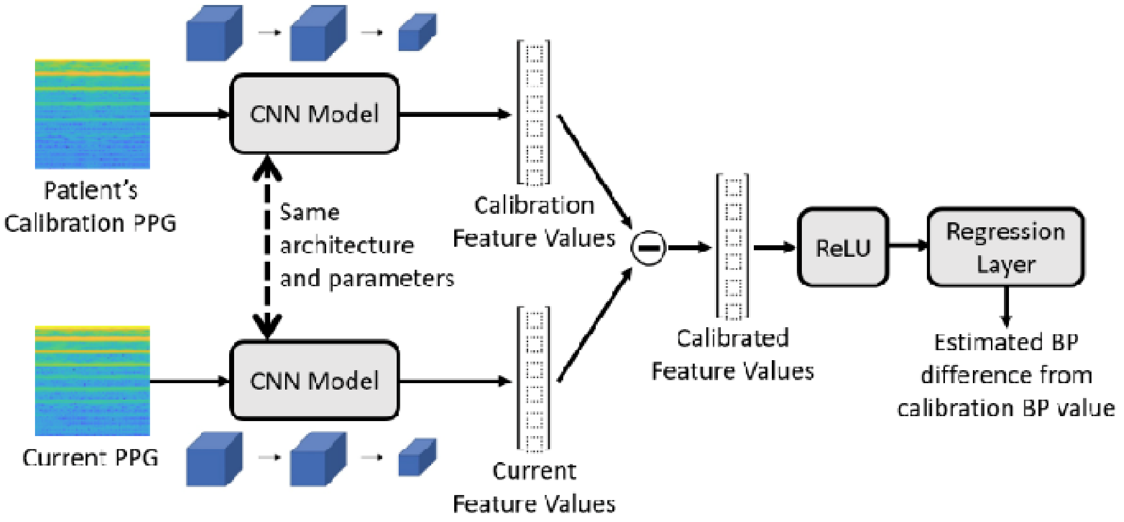
Noisy PPG and
ABP signals

[1] M. Saeed et al., "Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): a public-access intensive care unit database," Critical Care Medicine, vol. 39, no. 5, p. 952, 2011. https://archive.physionet.org/mimic2/

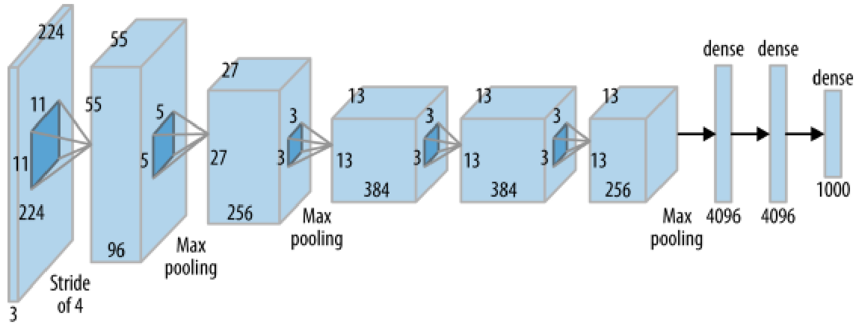
EPFL csem

Siamese network – Model architecture

- Using Siamese network
- Subnetworks working in parallel on two different inputs to compute feature vectors
  - Calibration segment – First available 30s window
  - Current PPG
- Compute difference between feature vectors
- Regression problem
  - Linear layer at the end
  - L1 loss function



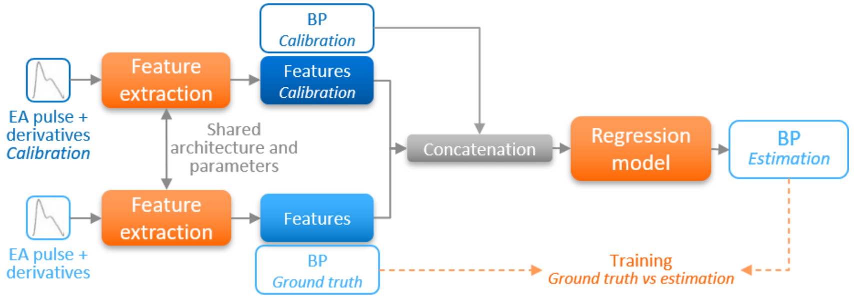
- **CNN architecture**
  - Inspired by AlexNet
  - Regularization with batch normalization

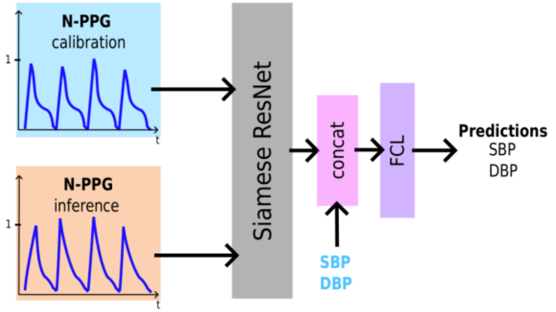


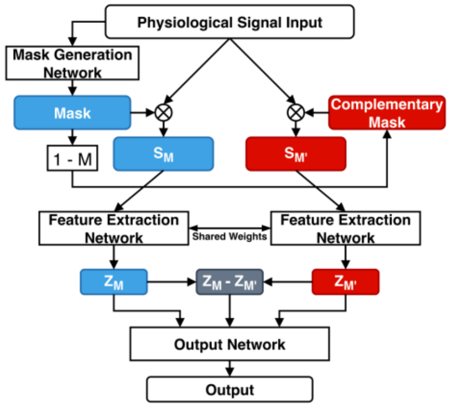
EPFL csem

Siamese network – similar approaches

- C. Aguet et al. - Feature learning for blood pressure estimation from photoplethysmography. EMBC 2021
  - Ensemble-averaged PPG pulses (EA pulse + derivatives)
  - Siamese with CNN backbone
  - Fusion and regression head: $[f_\theta(x_{est}), f_\theta(x_{cal}), BP_{cal}]$
- F.M. Dias et al. - Exploring the limitations of blood pressure estimation using the photoplethysmography signal. Physiological Measurement, 2025
  - 24s PPG segments
  - Siamese with ResNet backbone
  - Fusion and regression head: $[f_\theta(x_{inf}), f_\theta(x_{cal}), BP_{cal}]$
- W. Wang et al. - ΔBP-Net: Monitoring "Changes" in Blood Pressure Using PPG with Self-Contrastive Masking. IEEE JBHI, 2024
  - Detect clinically significant acute changes in SBP from PPG
  - Self-contrastive masking (SCM) creating 2 contrastive views of PPG input
  - ResNet with squeeze-and-excitation for feature extraction







Content

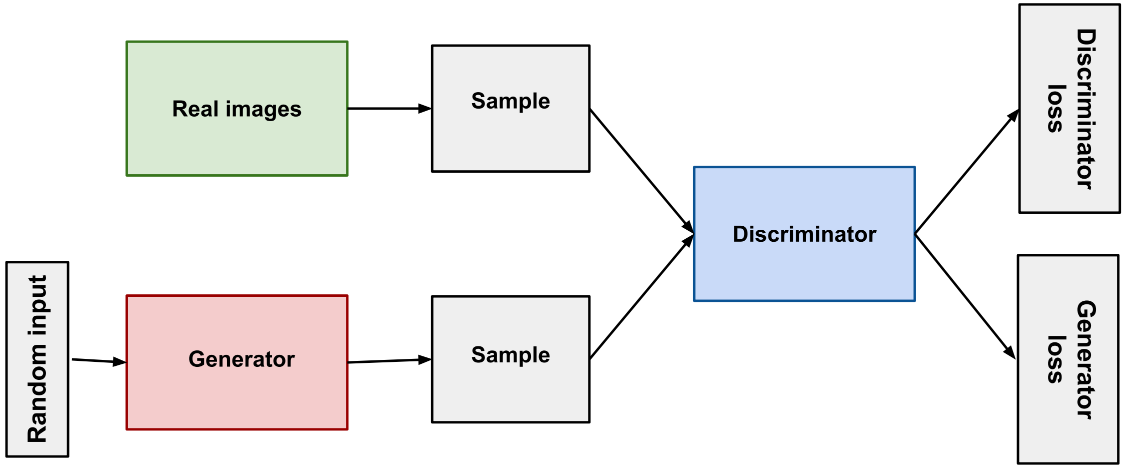
- Different architectures
  - Siamese neural network
  - Generative adversarial network
  - Explainable neural network

Generative adversarial network

- Proposed by Ian Goodfellow in 2014
- Generating new samples from scratch
- Learning by comparison

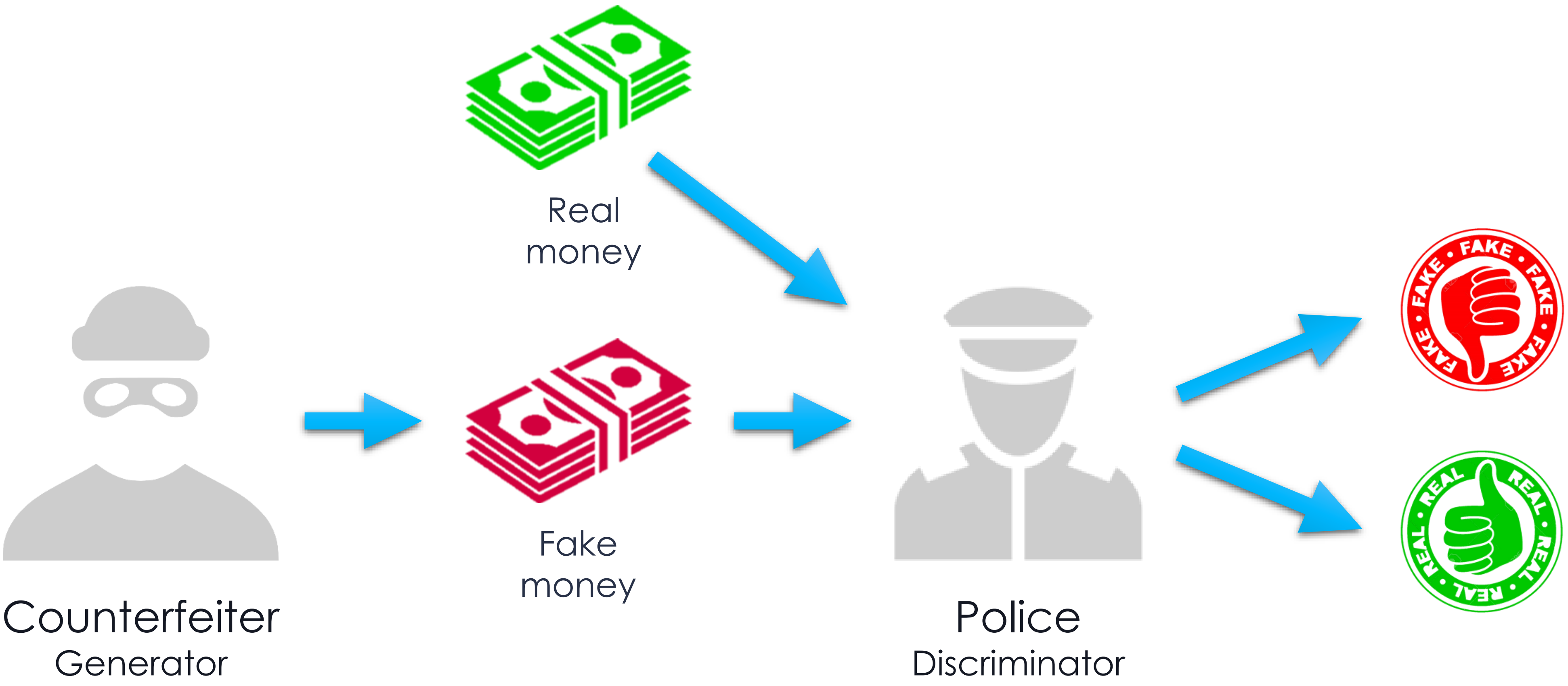
Training instability
Mode collapse
Sensitivity to architecture/parameters
Difficulty in sample quality evaluation

- **Generator**
  - Generate realistic-looking synthetic data
  - Try to fool discriminator
  - From noise to sample
  - Low to high dimensional space
- **Discriminator**
  - Distinguish generated and real samples
  - Binary classifier
  - From sample to decision
  - High to low dimensional space



EPFL csem

Generative adversarial network



EPFL csem

Generative adversarial network

- **GAN training**
  - Approximating data distribution and sampling from such approximation
  - Adversarial learning → Two network fight against each other
  - Simultaneous training → Alternating Generator and Discriminator updates

  - Discriminator
    - Maximize probability of assigning correct class → Maximize $logD(X)$

  - Generator
    - Maximize Discriminator uncertainty → Minimize $log\left(1 - D\left(G(X)\right)\right)$

$$min_G max_D \; \mathbb{E}_{X\sim\mu}[logD(X)] + \mathbb{E}_{X\sim\mu_Z}\left[log\left(1 - D\left(G(X)\right)\right)\right]$$

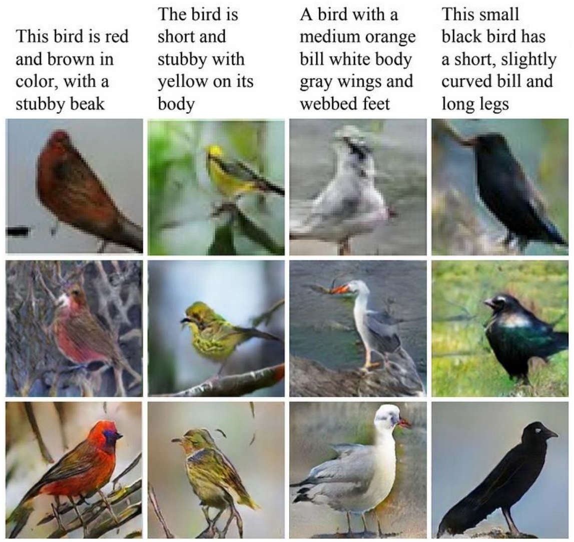
$$min_G max_D \; \mathbb{E}_{X\sim\mu}[logD(X)] + \mathbb{E}_{X\sim\mu_G}[log(1 - D(X))]$$

Generative adversarial network

- Applications

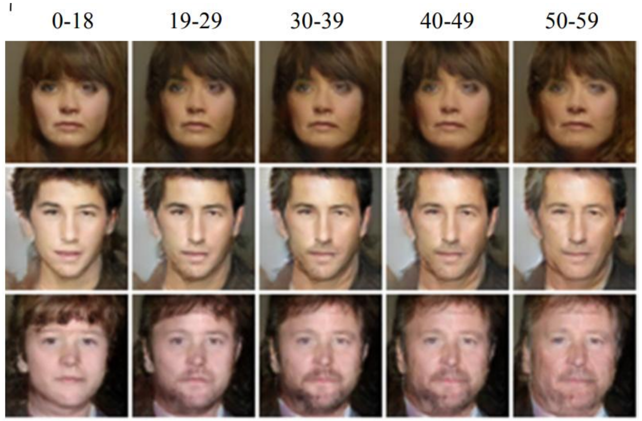
Image-to-image translation

Text-to-image synthesis



New poses generation

Face aging

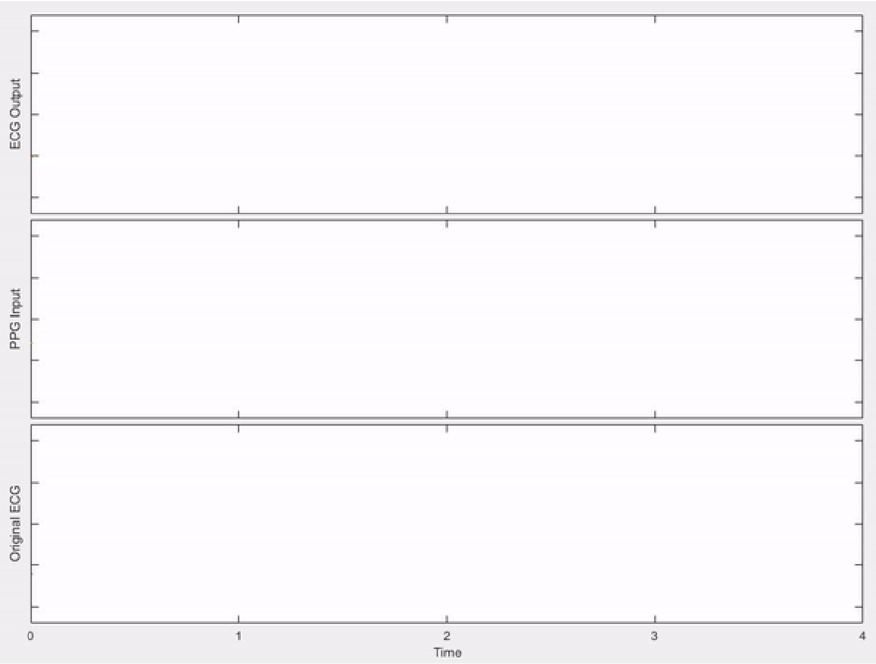


Generative adversarial network

Sarkar and Etemad 2020, CardioGAN: Attentive generative adversarial network with dual discriminators

AAAI conference on artificial intelligence

- GOAL
  - Enable ECG-level cardiac monitoring from PPG-only wearables
  - PPG common in wearables but lack ECG's richness
  - CardioGAN – PPG to ECG translation
  - Introduce attention-guided generation and joint time-frequency adversarial learning

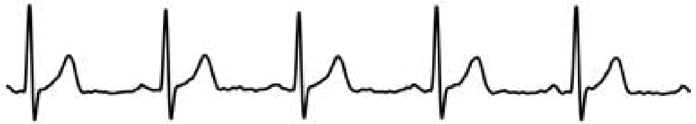


Generative adversarial network

- **Electrocardiogram** (ECG)
  - Electrical measurement of cardiac activity
  - Continuous health monitoring
  - Complicated integration into wearables
    → Non-continuous and sporadic measure

- **Photoplethysmogram** (PPG)
  - Optical measurement of volumetric changes in blood circulation
  - Simple, wearable-friendly, and low-cost



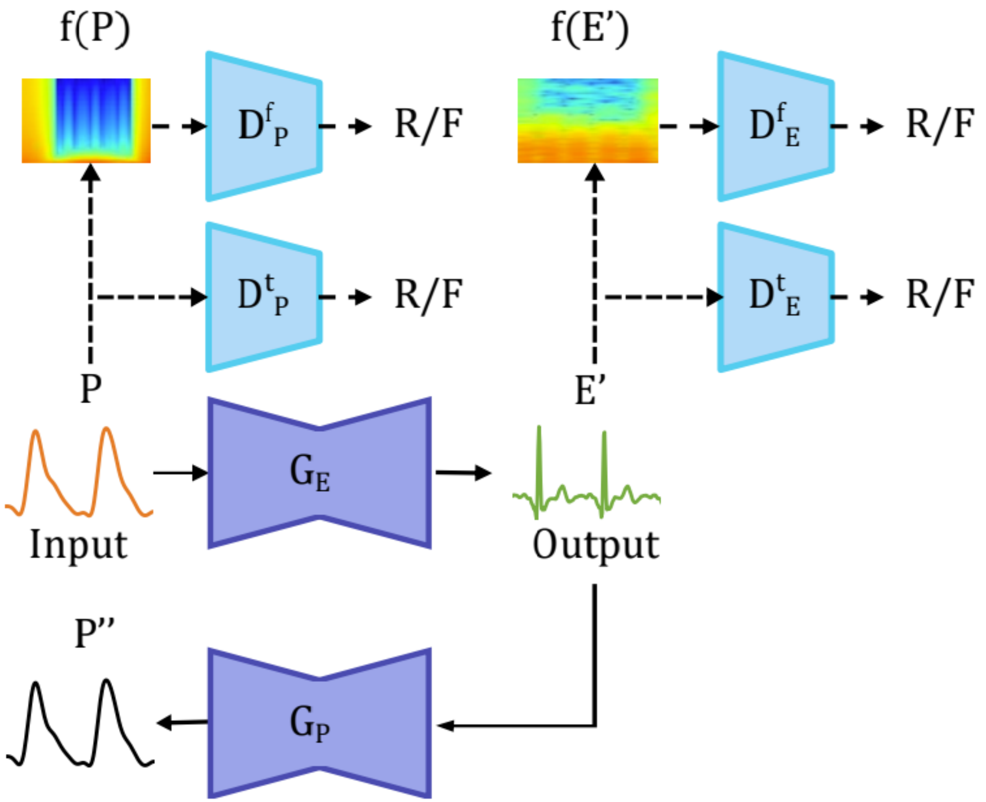

ECG


PPG

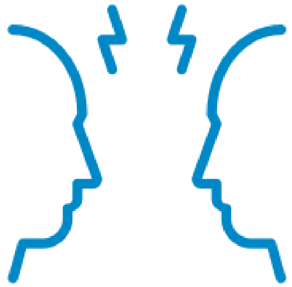
Generative adversarial network

Learn the mapping between PPG (P) and ECG (E)

- Generator (G)
  - CNN with attention mechanism
  - Forward mapping: $G_E: P \rightarrow E$
  - Fake ECG: $E' = G_E(P)$
  - Reconstructed PPG: $P'' = G_P\left(G_E(P)\right)$
- Discriminator (D)
  - Time-domain: $D_E^t: E\ vs\ E'$
  - Frequency-domain: $D_E^f: f(E)\ vs\ f(E')$
  - With $f(x) = STFT(x)$
  - Short-Time Fourier Transform



Generative adversarial network



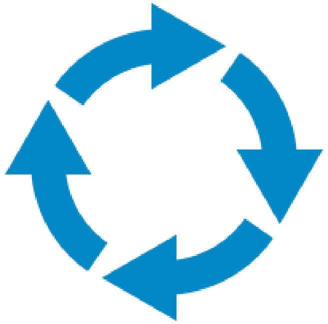
- **Adversarial loss**
  - Forward
    - Time-domain $\rightarrow L_{adv}(G_E, D_E^t) = E_{e\sim E}\left[log\left(D_E^t(e)\right)\right] + E_{p\sim P}\left[log\left(1 - D_E^t(G_E(p))\right)\right]$
    - Frequency-domain $\rightarrow L_{adv}(G_E, D_E^f) = E_{e\sim E}\left[log\left(D_E^f(f(e))\right)\right] + E_{p\sim P}\left[log\left(1 - D_E^f\left(f(G_E(p))\right)\right)\right]$
  - Backward
    - Time-domain $\rightarrow L_{adv}(G_P, D_P^t)$
    - Frequency-domain $\rightarrow L_{adv}(G_E, D_E^f)$

EPFL csem

Generative adversarial network

- **Cycle loss**

$$L_{cycle}(G_E, G_P) = E_{e\sim E}\left[\left\|G_E\big(G_P(e)\big) - e\right\|_1\right] + E_{p\sim P}\left[\left\|G_P\big(G_E(p)\big) - p\right\|_1\right]$$



  - Ensure forward and backward mappings are consistent
    - $p \rightarrow G_E(p) \rightarrow G_P\big(G_E(p)\big) \approx p$
    - $e \rightarrow G_P(e) \rightarrow G_E\big(G_P(e)\big) \approx e$

- **Final loss**

$$L_{CardioGAN} = \alpha L_{adv}(G_E, D_E^t) + \alpha L_{adv}(G_P, D_P^t) + \beta L_{adv}\big(G_E, D_E^f\big) + \beta L_{adv}\big(G_P, D_P^f\big) + \lambda L_{cycle}(G_E, G_P)$$

  - α and β are adversarial loss coefficients
  - λ cycle consistency loss coefficient

EPFL csem

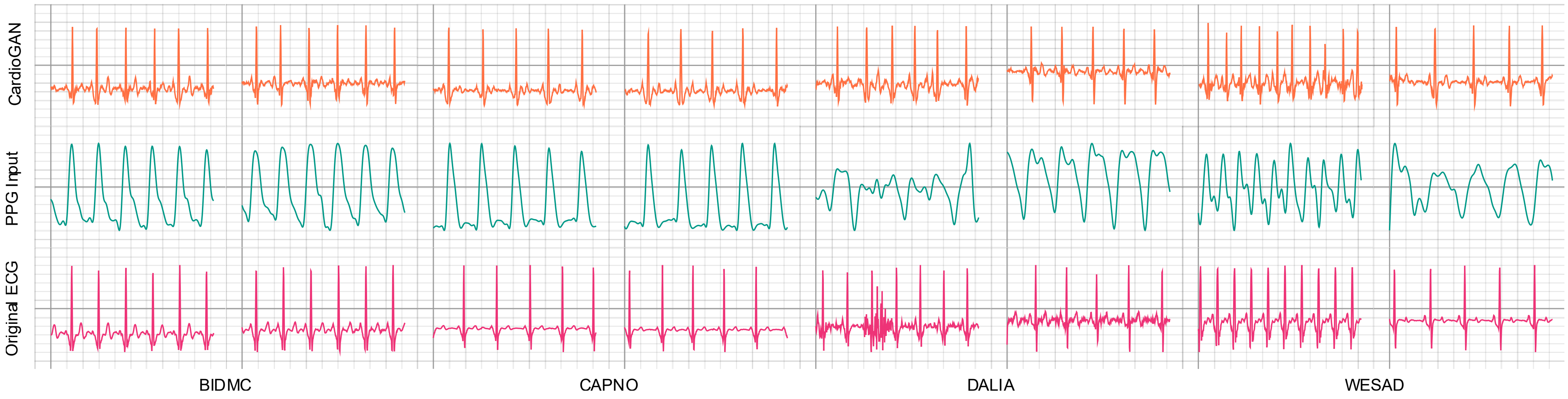
Generative adversarial network

- Popular ECG-PPG **datasets**
  - BIDMC database – 53 ICU subjects, 8min
  - CAPNO database – 42 subjects, 8mim
  - DALIA database – 15 subjects, ~2h during daily life activities
  - WESAD database – 15 subjects, >1h performing specific tasks

- EGC and PPG **preprocessing**
  1. Resampling to 128 Hz
  2. Filtering – band-pass FIR filter
  3. Z-score normalization on recording (zero mean and unit variance)
  4. Segmentation into 4s windows (512 data points)
  5. Min-max normalization on window [-1, 1]

EPFL csem

Generative adversarial network



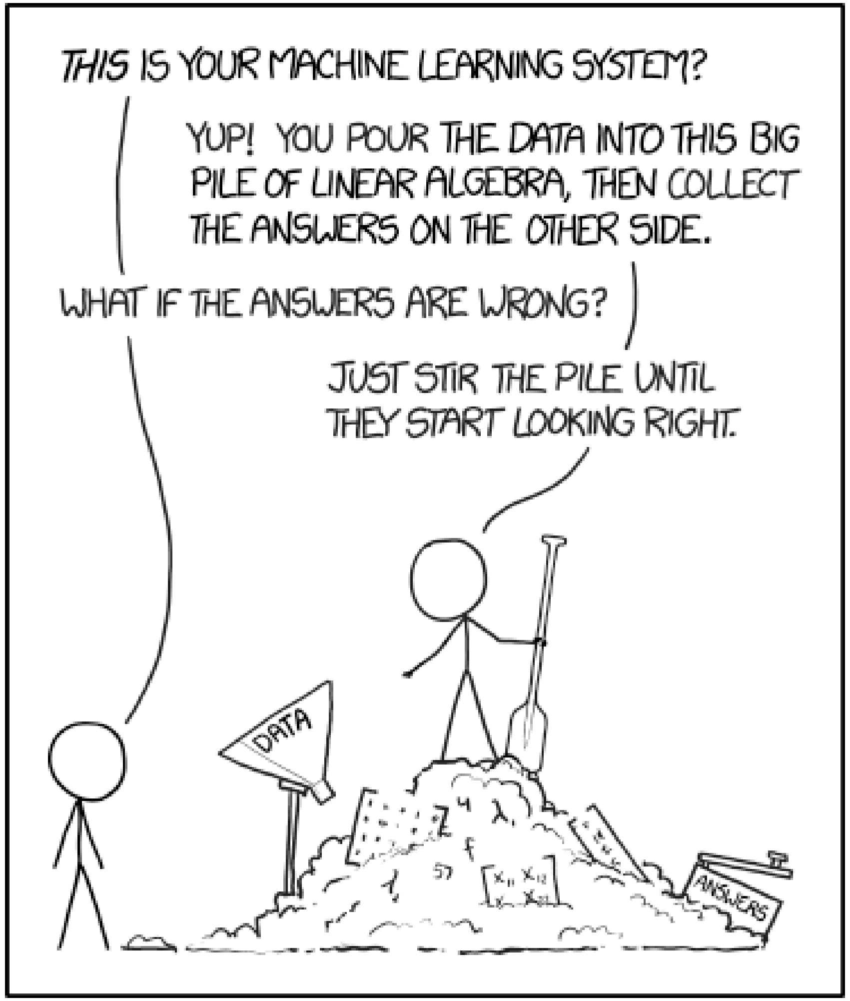
- Proposed novel framework for generating continuous ECG from PPG
- ECG generated by CardioGAN provides more reliable HR measurements compared to the original input PPG

- **Future direction**
  - Evaluate on other tasks: abnormal heart rhythms detection
  - Define more robust evaluation metrics

Content

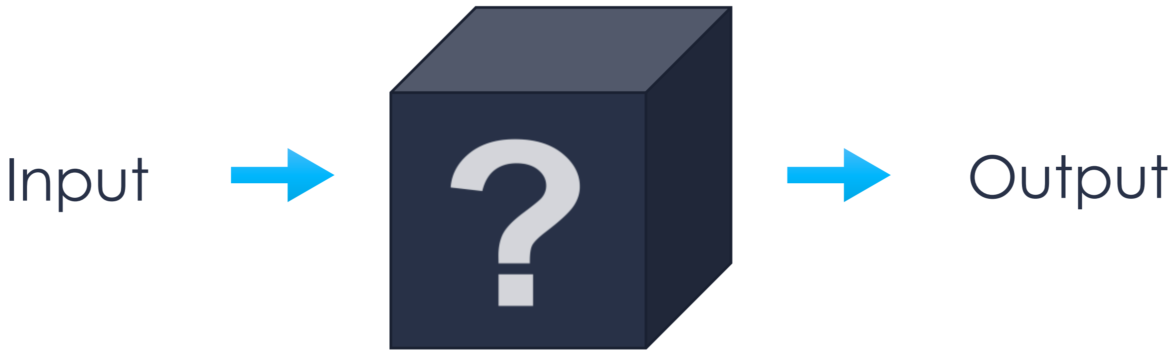
- Different architectures
  - Siamese neural network
  - Generative adversarial network
  - Explainable neural network

Explainable neural network



EPFL csem

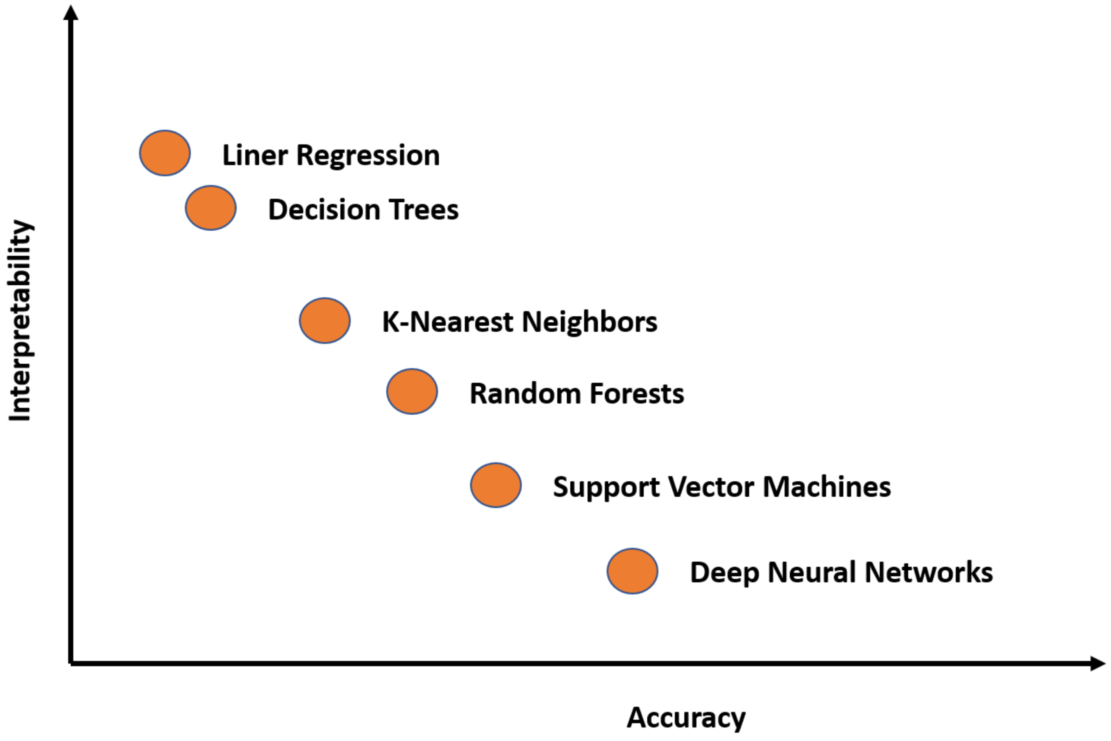
Explainable neural network



- Critics of ML → “black-box” model
- System that can produce valuable output, but which human might not understand
- But carefully constructed ML model can be verifiable and understandable
- Interpretable model = can tell how the model came to a decision
- In healthcare, an uninterpretable model can be unsafe or unusable

Explainable neural network

- Some ML methods are more interpretable than others
- Interpretability ≠ simplicity
- Linear regression, decision trees, generative additive models are inherently interpretable
- Interpretability often comes at the expense of power and accuracy
- Trade-off between interpretability and accuracy
- NN in low interpretability, high accuracy end of spectrum

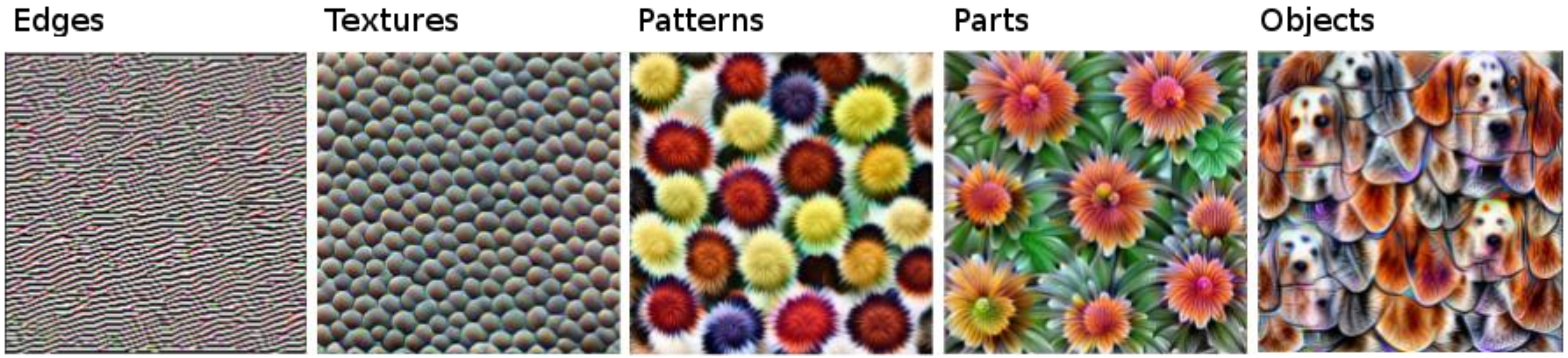


Explainable neural network

- **Intrinsic**
  - Explainability built into the model
  - Achieved by restricting the model complexity
  - Examples: linear regression, decision tree, etc.
- **Post-hoc**
  - Interpretable techniques help to reveal how models make predictions
  - After training $\rightarrow$ Does NOT change model only our understanding
  - Examples: permutation feature importance

Explainable neural network

- CNN for image classification
  - Visualize features to understand what the network is seeing
  - Each layer learns a specific feature
  - Higher layers learn more complex features based on simpler features learned by lower layers

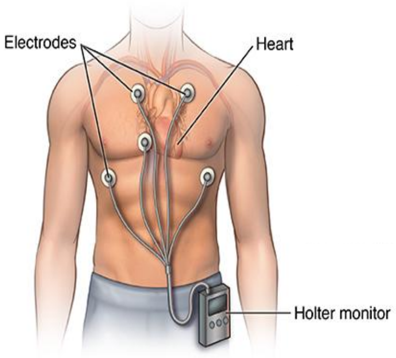


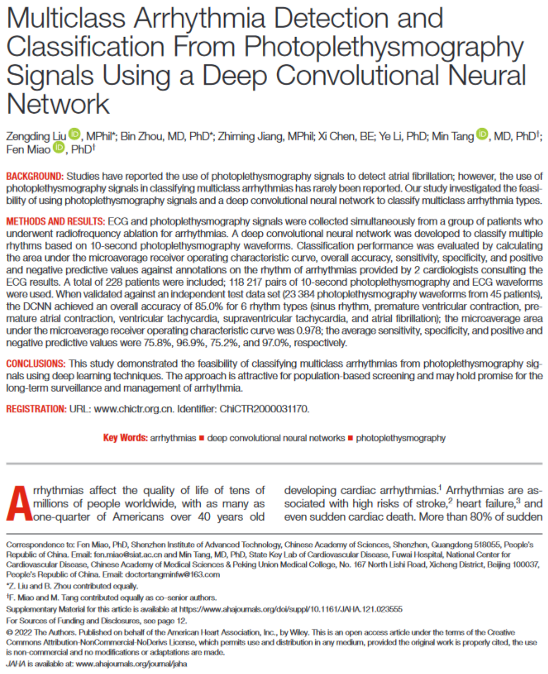
Explainable neural network

Liu et al. 2022, Multiclass arrhythmia detection and classification from photoplethysmography signals using a deep convolutional neural network

Journal of the American Heart Association

- GOAL
  - Classify multiclass arrhythmia types using PPG and deep CNN
  - Explainable tools to verify relevance of learned patterns





Multiclass Arrhythmia Detection and Classification From Photoplethysmography Signals Using a Deep Convolutional Neural Network

Zengding Liu, MPhil*; Bin Zhou, MD, PhD*; Zhiming Jiang, MPhil; Xi Chen, BE; Ye Li, PhD; Min Tang, MD, PhD†; Fen Miao, PhD†

BACKGROUND: Studies have reported the use of photoplethysmography signals to detect atrial fibrillation; however, the use of photoplethysmography signals in classifying multiclass arrhythmias has rarely been reported. Our study investigated the feasibility of using photoplethysmography signals and a deep convolutional neural network to classify multiclass arrhythmia types.

METHODS AND RESULTS: ECG and photoplethysmography signals were collected simultaneously from a group of patients who underwent radiofrequency ablation for arrhythmias. A deep convolutional neural network was developed to classify multiple rhythms based on 10-second photoplethysmography waveforms. Classification performance was evaluated by calculating the area under the microaverage receiver operating characteristic curve, overall accuracy, sensitivity, specificity, and positive and negative predictive values against annotations on the rhythm of arrhythmias provided by 2 cardiologists consulting the ECG results. A total of 228 patients were included; 118 217 pairs of 10-second photoplethysmography and ECG waveforms were used. When validated against an independent test data set (23 384 photoplethysmography waveforms from 45 patients), the DCNN achieved an overall accuracy of 85.0% for 6 rhythm types (sinus rhythm, premature ventricular contraction, premature atrial contraction, ventricular tachycardia, supraventricular tachycardia, and atrial fibrillation); the microaverage area under the microaverage receiver operating characteristic curve was 0.978; the average sensitivity, specificity, and positive and negative predictive values were 75.8%, 96.9%, 75.2%, and 97.0%, respectively.

CONCLUSIONS: This study demonstrated the feasibility of classifying multiclass arrhythmias from photoplethysmography signals using deep learning techniques. The approach is attractive for population-based screening and may hold promise for the long-term surveillance and management of arrhythmia.

REGISTRATION: URL: www.chictr.org.cn. Identifier: ChiCTR2000031170.

Key Words: arrhythmias ■ deep convolutional neural networks ■ photoplethysmography

Arrhythmias affect the quality of life of tens of millions of people worldwide, with as many as one-quarter of Americans over 40 years old developing cardiac arrhythmias.[1] Arrhythmias are associated with high risks of stroke,[2] heart failure,[3] and even sudden cardiac death. More than 80% of sudden

Correspondence to: Fen Miao, PhD, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong 518055, People's Republic of China. Email: fen.miao@siat.ac.cn and Min Tang, MD, PhD, State Key Lab of Cardiovascular Disease, Fuwai Hospital, National Center for Cardiovascular Disease, Chinese Academy of Medical Sciences & Peking Union Medical College, No. 167 North Lishi Road, Xicheng District, Beijing 100037, People's Republic of China. Email: doctortangminfw@163.com

*Z. Liu and B. Zhou contributed equally.

†F. Miao and M. Tang contributed equally as co-senior authors.

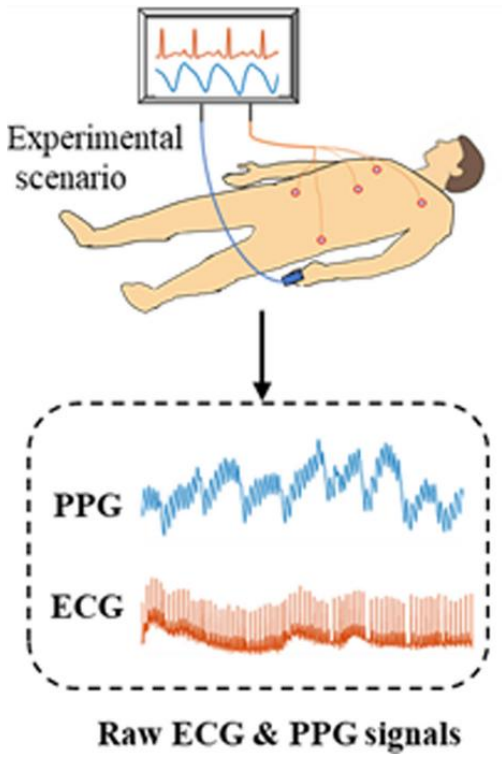
Supplementary Material for this article is available at https://www.ahajournals.org/doi/suppl/10.1161/JAHA.121.023555

For Sources of Funding and Disclosures, see page 12.

JAHA is available at: www.ahajournals.org/journal/jaha

Explainable neural network

- DATA
  - 242 patients with arrhythmia receiving radiofrequency catheter ablation
  - At Fuwai Hospital, Chinese Academy of Medical Sciences
  - Fingertip **PPG**
  - 3-lead **ECG** for reference



Explainable neural network

- Preprocessing
  1. Signal downsampling (250 to100 Hz)
  2. Denoising with bandpass filtering
     - ECG from 0.5 to 50 Hz
     - PPG from 0.5 to 10 Hz
  3. Segmentation
     - Split into 10s non-overlapping segments
  4. Segment normalization
  5. Signal quality exclusion
     - Remove segments with artifacts in PPG
     - Remove segments with noisy ECG reference
  6. Labels
     - Annotation of ECG segments from 2 cardiologists
     - Create labels according to ECG annotations

118’217 labeled 10s PPG segments from 228 patients

EPFL csem

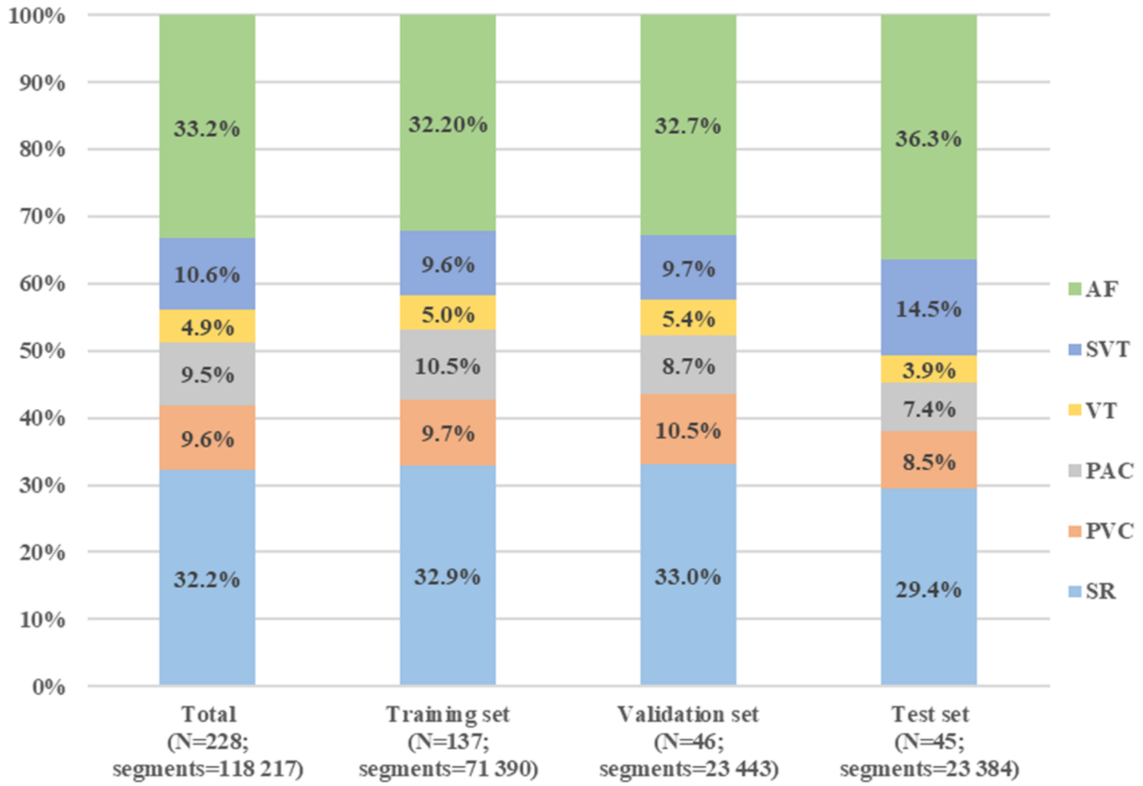
Explainable neural network

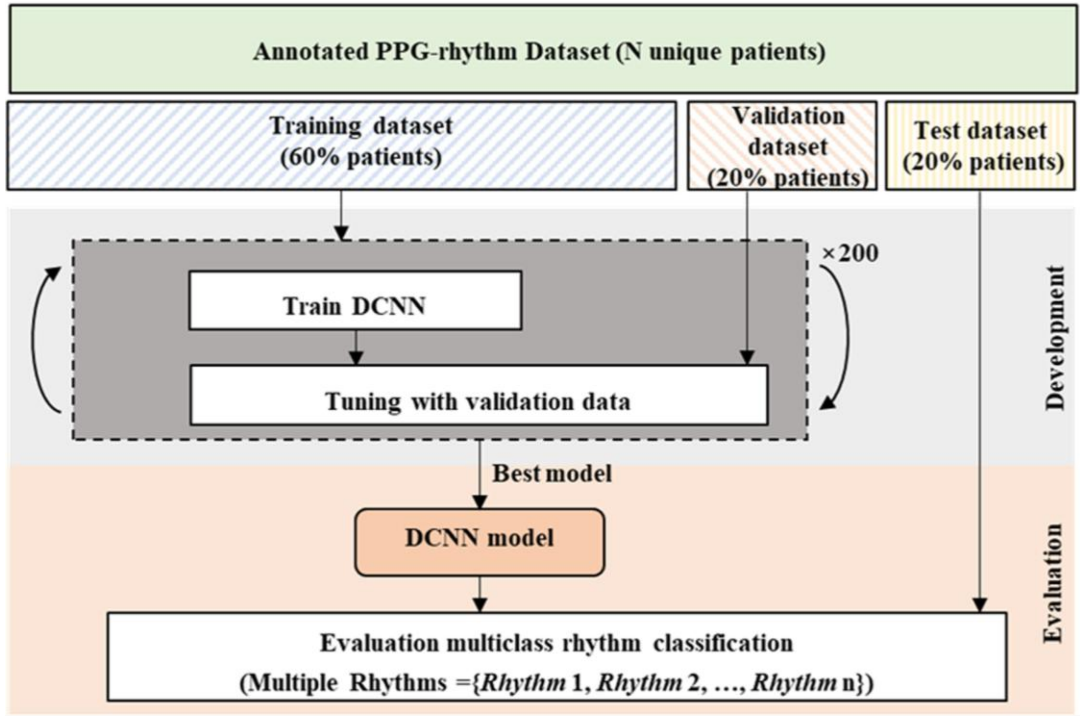
- Each segment has only 1 identified rhythm type
- Strong class imbalance

Rhythm type	**Number of segments**
Sinus rhythm	38081
Premature ventricular contraction (PVC)	11372
Premature atrial contraction (PAC)	11248
Ventricular tachycardia (VT)	5783
Supraventricular tachycardia (SVT)	12539
Atrial fibrillation (AF)	39194

Explainable neural network

- Split data
- 60% of patients in training set
- 20% of patients in validation set
- 20% of patients in test set

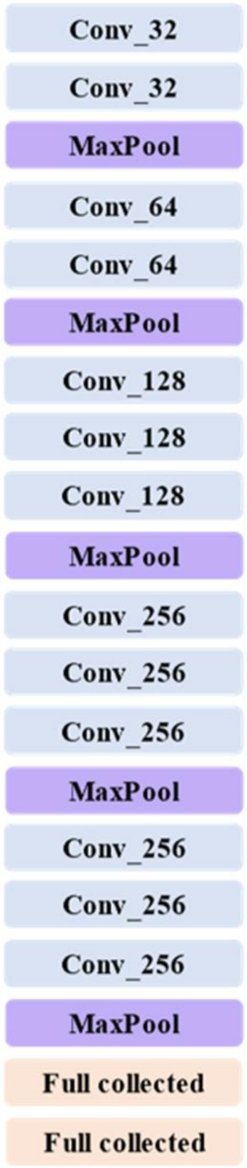




EPFL csem

Explainable neural network

- **Multiclass rhythm classification from PPG**
  - Model
    - Based on VGGNet-16 (deep CNN)
    - Adapted for 1-dimensional input signal
    - 13 convolutional layers
    - 5 max-pooling layers
    - 3 fully connected layers
  - Input: 10-second PPG segment
  - Output: label prediction of one of the rhythm type

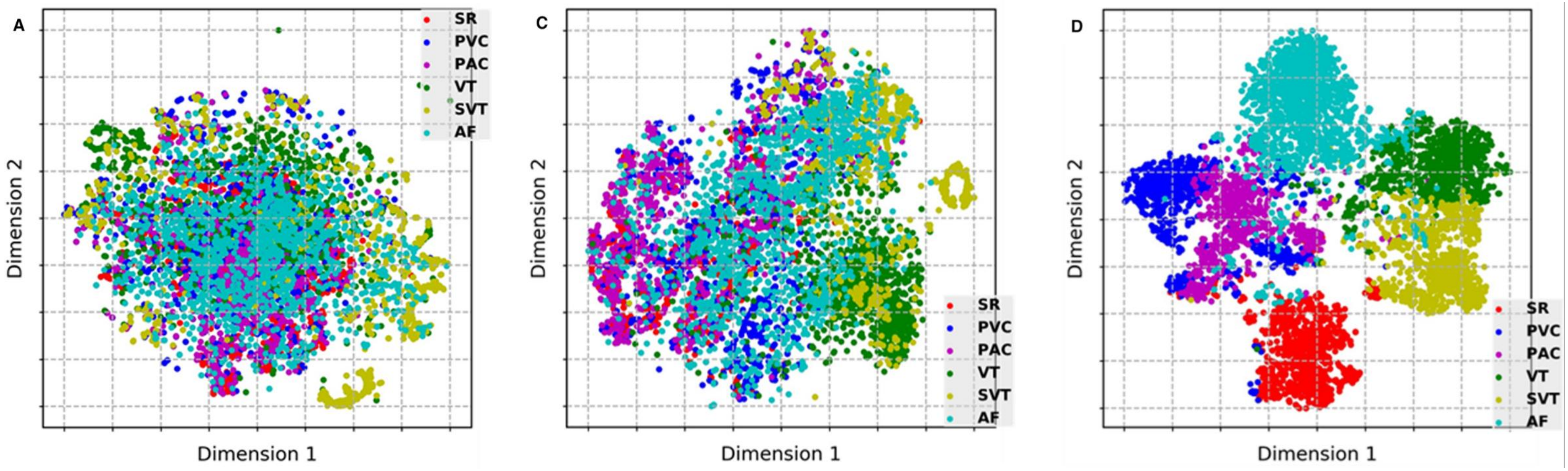


Explainable neural network

- **Visualization of proposed DCNN**
  - Improve understanding of model decision
  - Highlight regions in PPG important for the model prediction
  - t-distributed stochastic neighbor embedding (t-SNE)
    - Non-linear dimensionality reduction technique
    - Embedding high-dimensional data for visualization in low dimensional space
  - Guided gradient-weighted class activation mapping (Grad-CAM)
    - Fine-grained guided backpropagation
    - Gradient-weighted class activation mapping
    - Create high-resolution class-discriminative heatmap from final convolutional layer

Explainable neural network

- **t-distributed stochastic neighbor embedding**



Layer 2 | Layer 7 | Layer 13

EPFL csem

Explainable neural network

Sinus rhythm

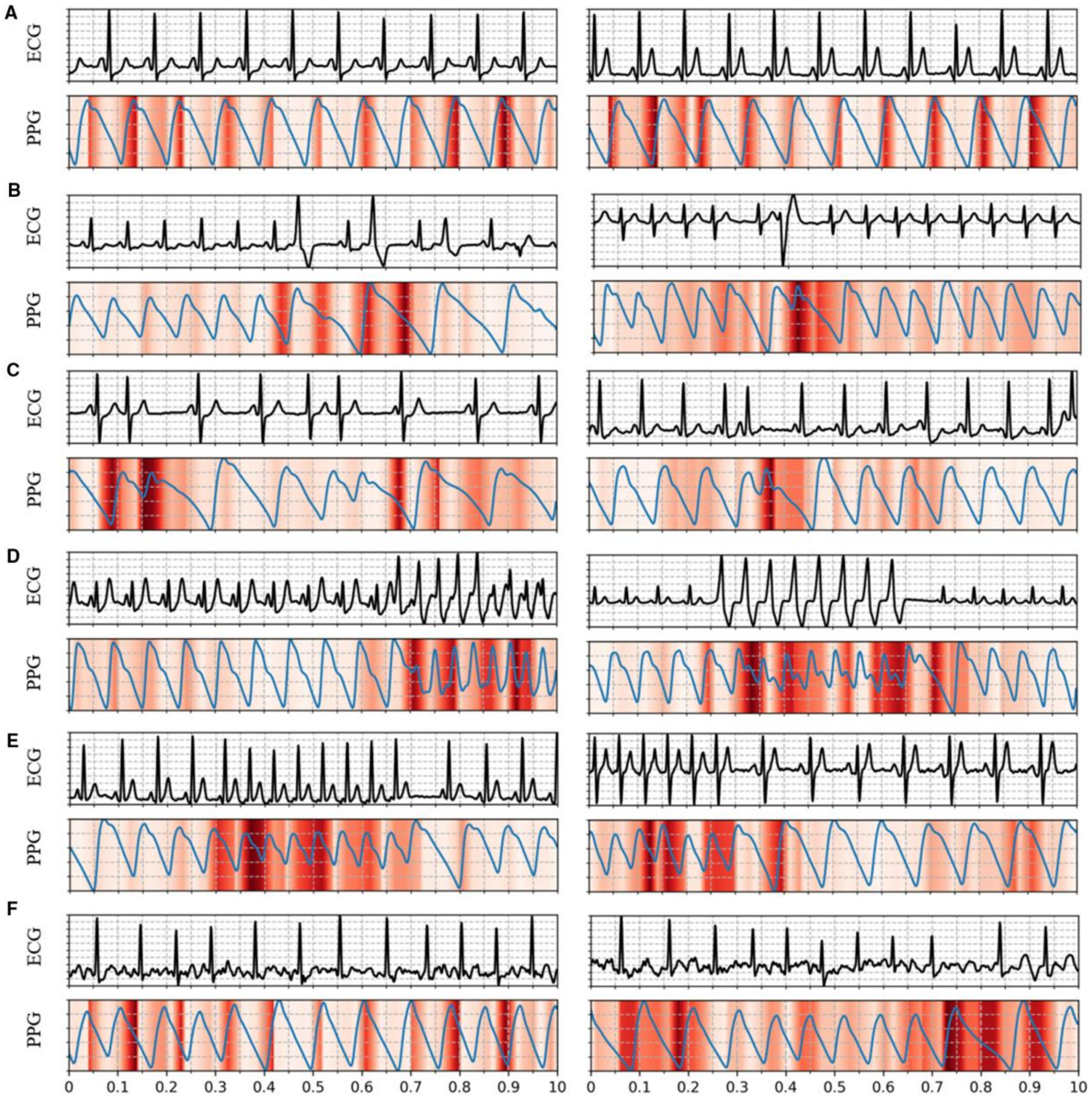
PVC
Premature ventricular contraction

PAC
Premature atrial contraction

VT
Ventricular tachycardia

SVT
Supraventricular tachycardia

AF
Atrial fibrillation



EPFL csem