The Coding-workshop will start soon!

Please MUTE your MIC and DISABLE your CAMERA!

The session will be

recorded

Coding AI

Introduction to PyTorch

By Kevin Trebing

About me

- Education
  - 2013-2017: B.Sc. Cognitive Science in Osnabrück, Germany
  - 2018-2020: M.Sc. Artificial Intelligence in Maastricht, Netherlands
- Worked on several AI-projects in different companies:
  - Predictive Maintenance
  - Weather prediction
  - Evolutionary game design
  - Automatic menu-card classification
  - … more
- Python-lover
- Extensive use of PyTorch in Master’s degree

What is PyTorch?

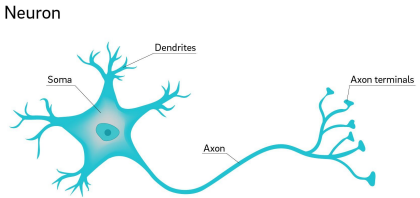
- Python library with C++ backend
- A library to replace NumPy in order to use GPUs
  - Similar functions
  - Similar syntax
- Deep learning library
  - Easy implementation of neural networks
  - Easy training of neural networks
  - Fully customizable (more advanced)
- Good intro resource: Deep Learning with PyTorch

Why pytorch?

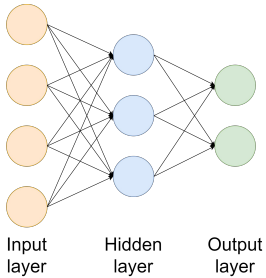
- Very Python-like:
  - Classes
  - If-else conditions
  - Loops
- Big and great community
- A lot of users → easy to get help (Stackoverflow, Forums)
- Gets used by many researchers → newest models available
- Good documentation (looking at you Tensorflow!)

(Artificial) Neural network - recap

- Not real neurons (just the idea)

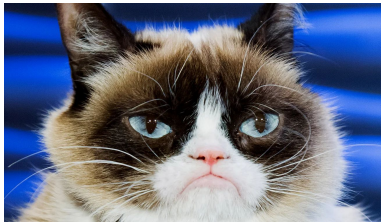


- **Artificial** neurons are values (weights) that are adapted
- Combining multiple neurons results in a neural network
  - E.g: Multilayer Perceptron (MLP)



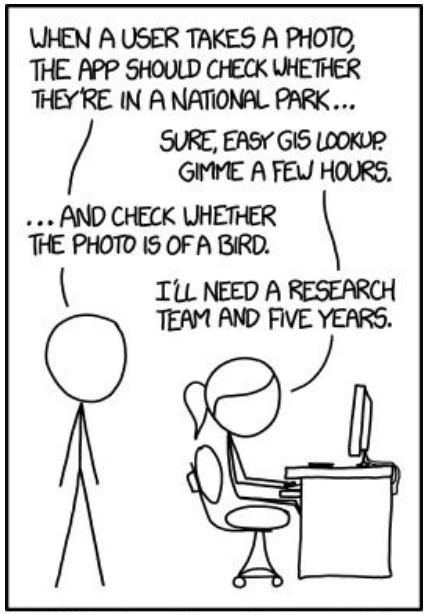
How to train a neural network? - dataset

Suppose we have examples:

Image				
Label	Dog	Dog	Cat	Cat

Very hard task already!

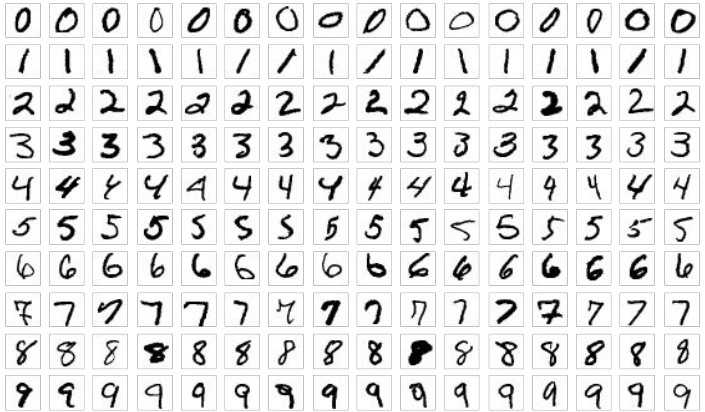
Relevant xkcd



How to train a neural network? - Start easy

Recognize handwritten digits

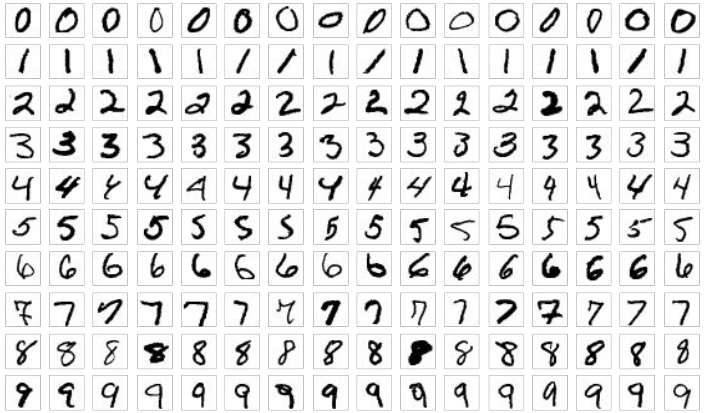
- MNIST-dataset:
  - Input: 28x28 pixel images
  - Output: Label



How to train a neural network? - Start easy

Recognize handwritten digits

- MNIST-dataset:
  - Input: 28x28 pixel images
  - Output: Label



How to train a neural network? - Normalize data

- Data should be normalized -> In the range between 0 and 1 or -1 and 1
- Reduces chance of saturated neurons -> unsaturated is better for learning

How to train a neural network? - Normalize data

- Data should be normalized -> In the range between 0 and 1 or -1 and 1
- Reduces chance of saturated neurons -> unsaturated is better for learning

- Can be done by calculating mean and standard deviation:

$$\frac{X-\mu_x}{\sigma_x}$$

- Or min-max scaling:

$$\frac{X-min(x)}{max(x)-min(x)}$$

How to train a neural network? - Loss functions

How to determine how good/bad a model’s prediction is?

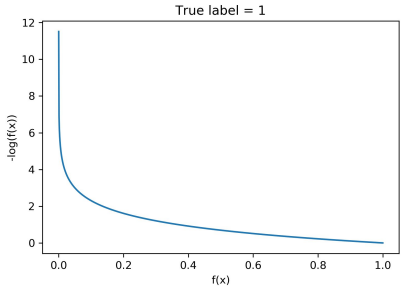
-> Use a function that compares the true label and predicted one

How to train a neural network? - Loss functions

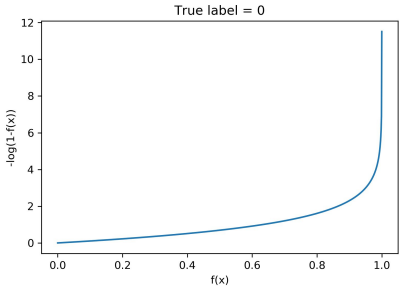
How to determine how good/bad a model's prediction is?

-> Use a function that compares the true label and predicted one

- Binary classification
  - Log-loss



$$-log(f(x))$$



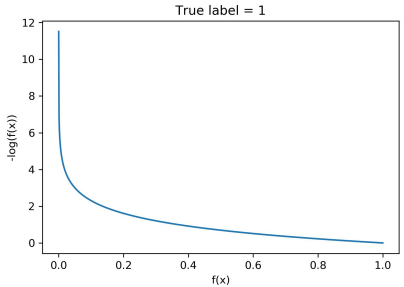
$$-log(1-f(x))$$

How to train a neural network? - Loss functions

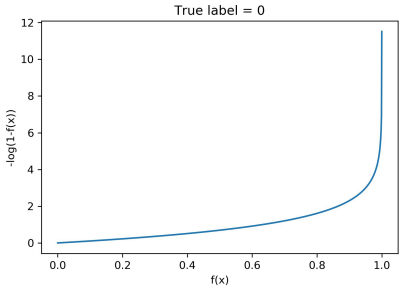
How to determine how good/bad a model's prediction is?

-> Use a function that compares the true label and predicted one

- Binary classification
  - Log-loss
- More classes
  - Cross-entropy-loss
- Regression task
  - L1 loss $|y_{pred} - y_{true}|$
  - L2 loss $|y_{pred} - y_{true}|^2$



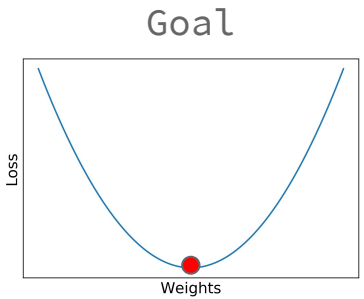
$-log(f(x))$



$-log(1 - f(x))$

How to train a neural network? - Optimizers

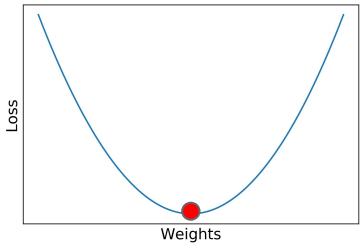
Minimize loss and adapt weights with backpropagation -> “optimize”



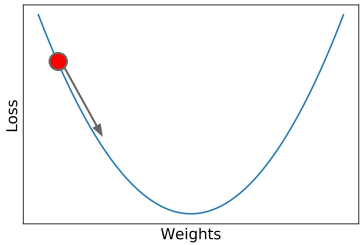
How to train a neural network? - Optimizers

Minimize loss and adapt weights with backpropagation -> “optimize”

Goal



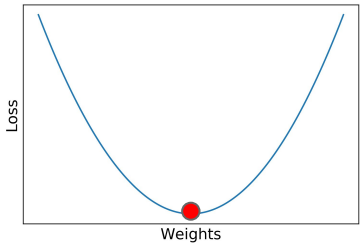
Idea



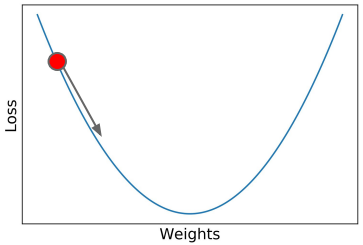
How to train a neural network? - Optimizers

Minimize loss and adapt weights with backpropagation -> “optimize”

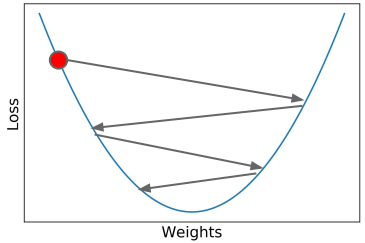
Goal



Idea



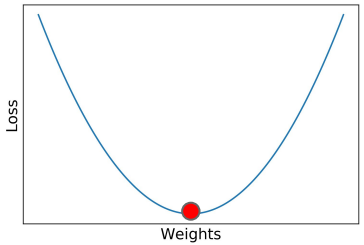
Reality 1



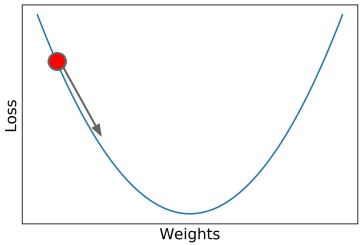
How to train a neural network? - Optimizers

Minimize loss and adapt weights with backpropagation -> “optimize”

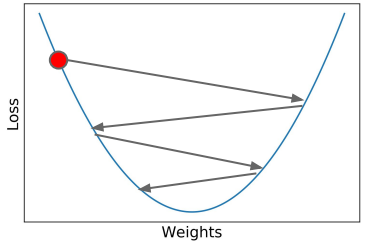
Goal



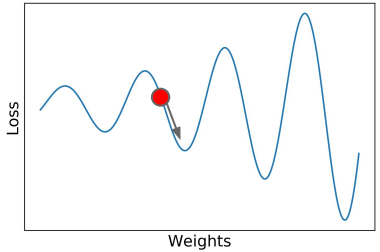
Idea



Reality 1

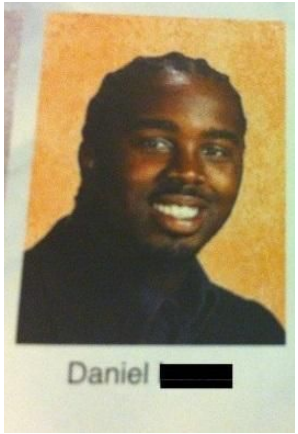


Reality 2

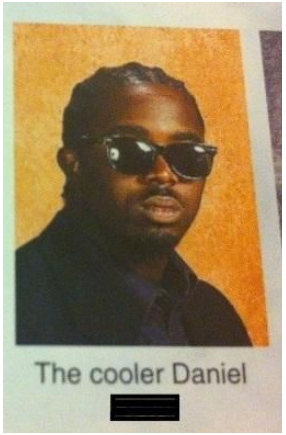


Important parameters: **optimizer** and **learning rate**

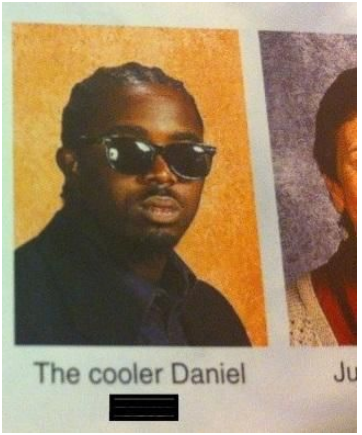
Choose your optimizer



SGD



RMSprop



Adam ...

Optimizer receives the model’s parameter and learning rate

Summary - What do we need?

1. Dataset
   a. Train
   b. Validation
   c. Test

   from torch.utils.data import Dataset, Dataloader
2. Neural network model

   from torch import nn
3. Loss function

   from torch.nn import CrossEntropyLoss

   Or from torch.nn.functional import cross_entropy
4. Optimizer

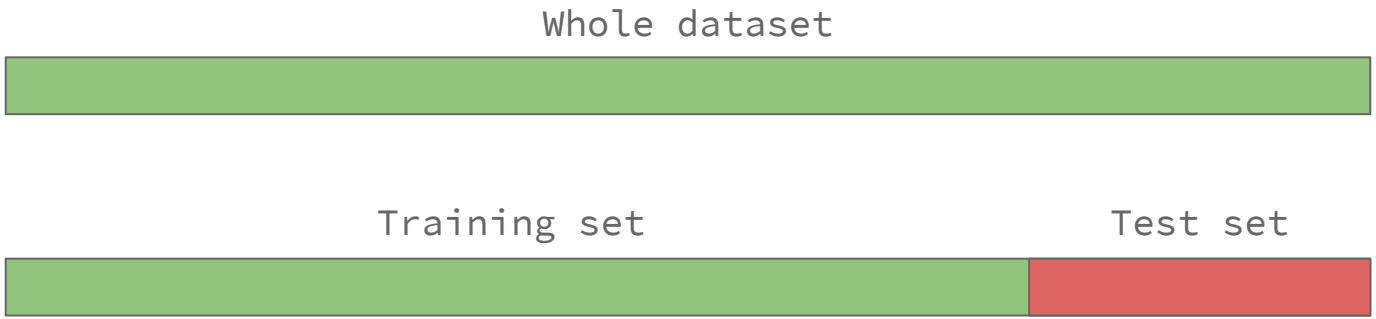
   from torch.optim import Adam, RMSprop, SGD
5. Hyperparameters:
   a. How long to train - epochs
   b. Batch size
   c. Learning rate

Dataset - closer look

Train network to generalize -> able to predict/classify unseen samples

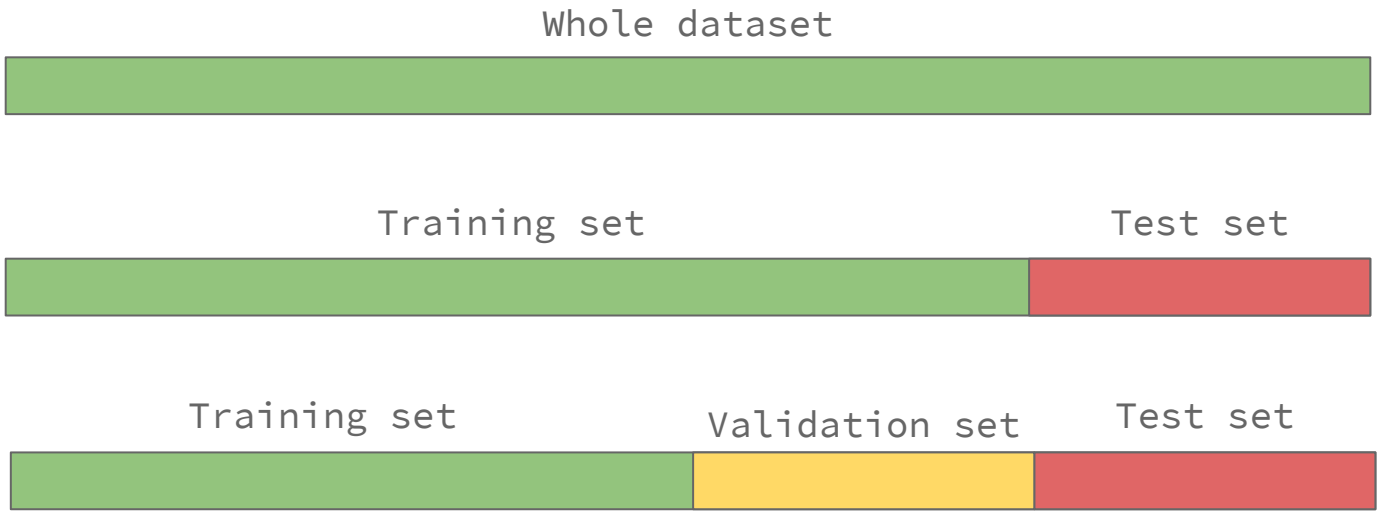
Dataset - closer look

Train network to generalize -> able to predict/classify unseen samples

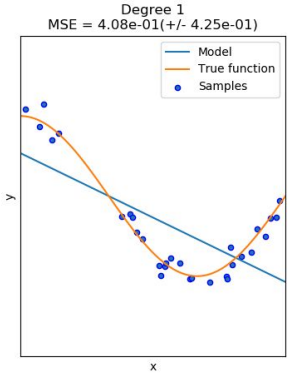


Dataset - closer look

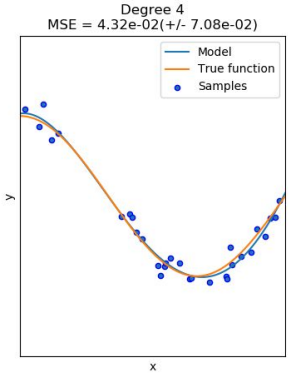
Train network to generalize -> able to predict/classify unseen samples



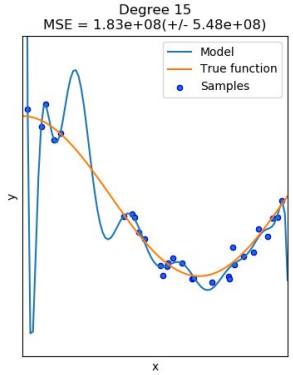
Why the hassle?



Underfit

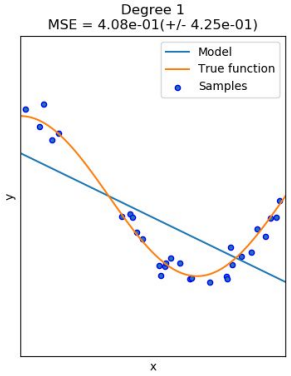


Good fit

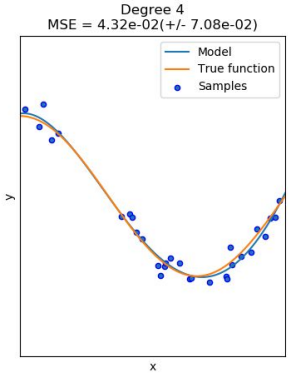


Overfit

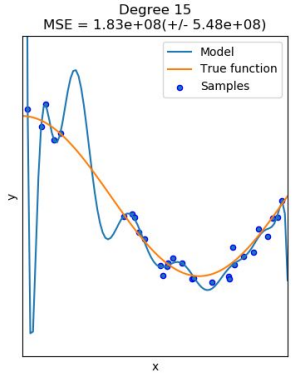
Why the hassle?



Underfit

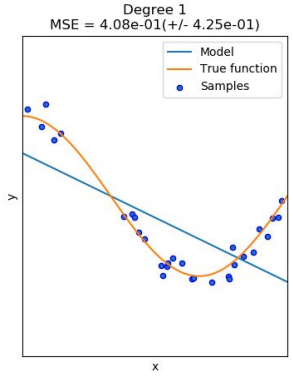


Good fit

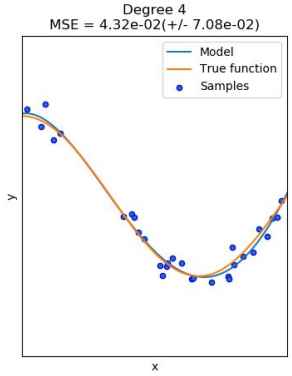


Overfit

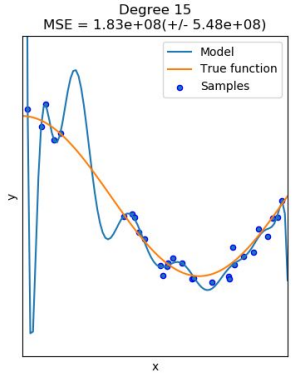
Why the hassle?



Underfit

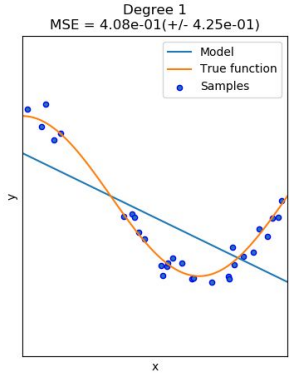


Good fit

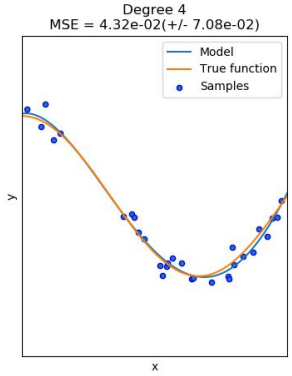


Overfit

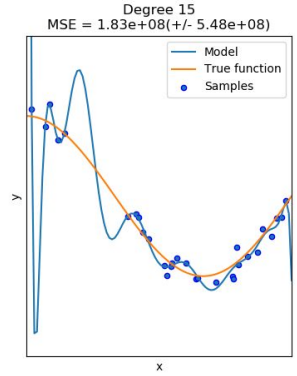
Why the hassle?



Underfit



Good fit



Overfit

Summary (again) - What do we need?

1. Dataset
    a. Train
    b. Validation
    c. Test

   from torch.utils.data import Dataset, Dataloader
   from sklearn.model_selection import train_test_split
   Or from torch.utils.data import random_split

2. Neural network model

   from torch import nn

3. Loss function

   from torch.nn import CrossEntropyLoss
   Or from torch.nn.functional import cross_entropy

4. Optimizer

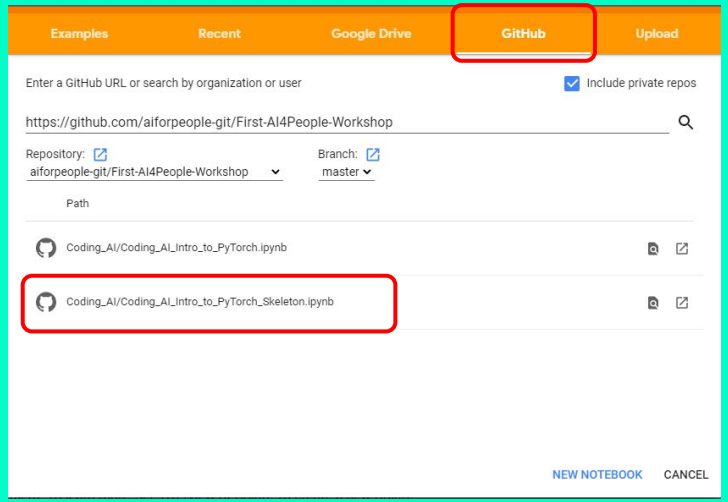
   from torch.optim import Adam, RMSprop, SGD

5. Hyperparameters:
    a. How long to train - epochs
    b. Batch size
    c. Learning rate

Now to the coding!

An example of how to write code to train a neural network with PyTorch

1. Go to: https://colab.research.google.com/
2. Open the code-skeleton:

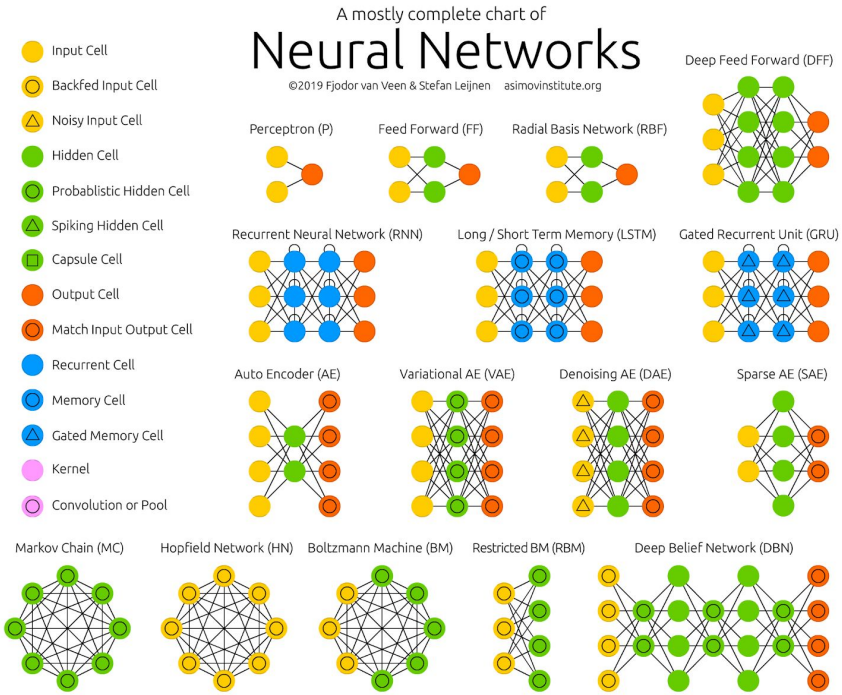


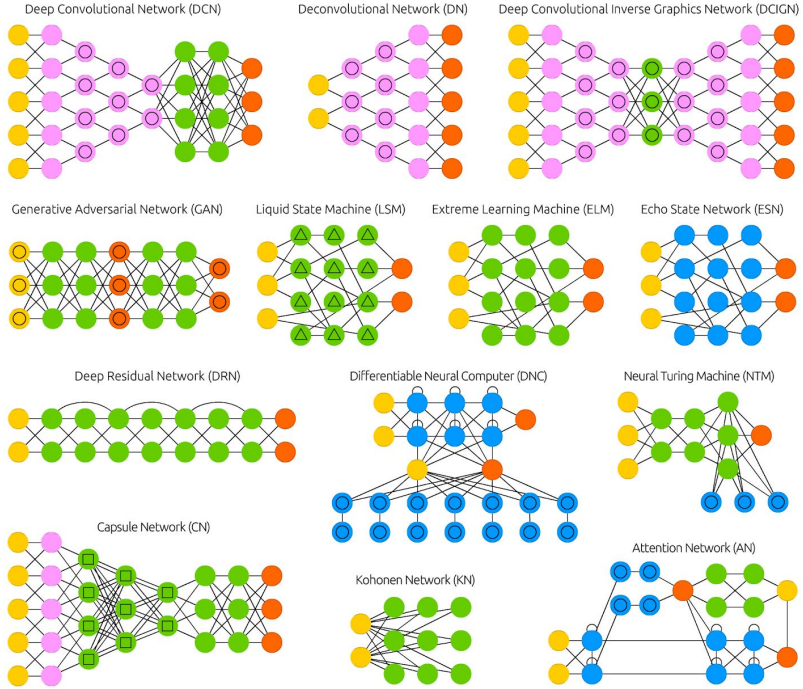
Thank you for listening! Questions?

Advanced pytorch things

- Learning rate scheduler — from torch.optim.lr_scheduler
- Tensorboard logging — from torch.utils.tensorboard import SummaryWriter
- Saving the trained model — torch.save({'state_dict': model.state_dict}, 'saved_model.pt')
- Loading the saved model — model.state_dict = torch.load('saved_model.pt')['state_dict']

Oh, the possibilities!





Taken from: https://www.asimovinstitute.org/neural-network-zoo/

References

Axon: https://medicalxpress.com/news/2018-07-neuron-axons-spindly-theyre-optimizing.html

Cat image 1: https://timesofindia.indiatimes.com/life-style/relationships/pets/5-things-that-scare-and-stress-your-cat/articleshow/67586673.cms

Cat image 2: https://www.wired.com/story/grumpy-cat-obit/

XKCD: https://xkcd.com/1425/

MNIST: https://en.wikipedia.org/wiki/MNIST_database

Over and underfit: https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html