Since 2004
UET
ĐẠI HỌC CÔNG NGHỆ, ĐHQGHN
VNU-University of Engineering and Technology

Since 1906
VNU
ĐẠI HỌC QUỐC GIA HÀ NỘI
Vietnam National University, Hanoi

INT3405 - Machine Learning

Lecture 11: Deep Learning - P2

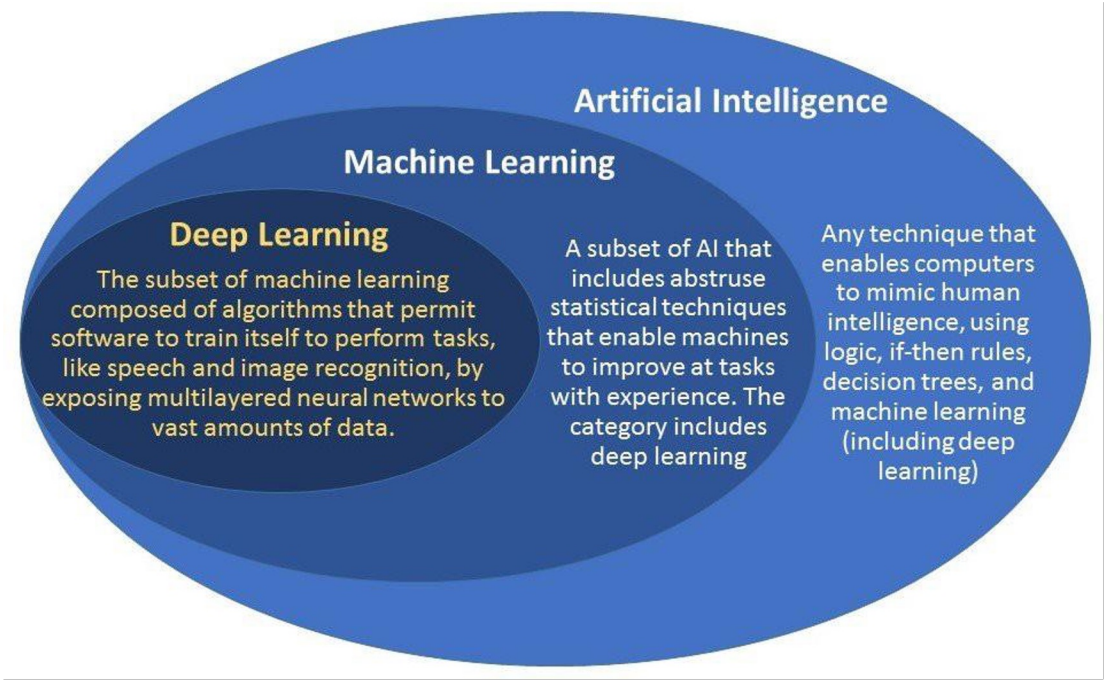
Duc-Trong Le & Viet-Cuong Ta

Hanoi, 11/2023

Outline

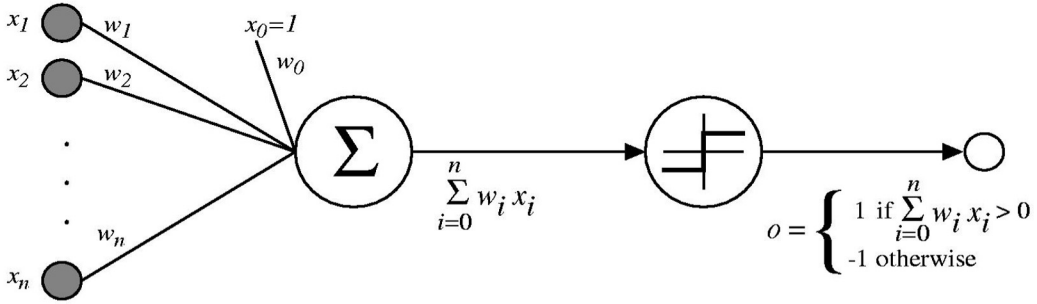
- Typical Architectures:
  - Convolutional Neural Networks
  - Recurrent Neural Networks
  - Attention Mechanism
- Overfitting Reduction
  - Dropout, Batch Norm
  - Early Stopping
  - Cross Validation

Recap: Machine Learning vs. Deep Learning



https://lerablog.org/technology/ai-artificial-intelligence-vs-machine-learning-vs-deep-learning/

Recap: Perceptron

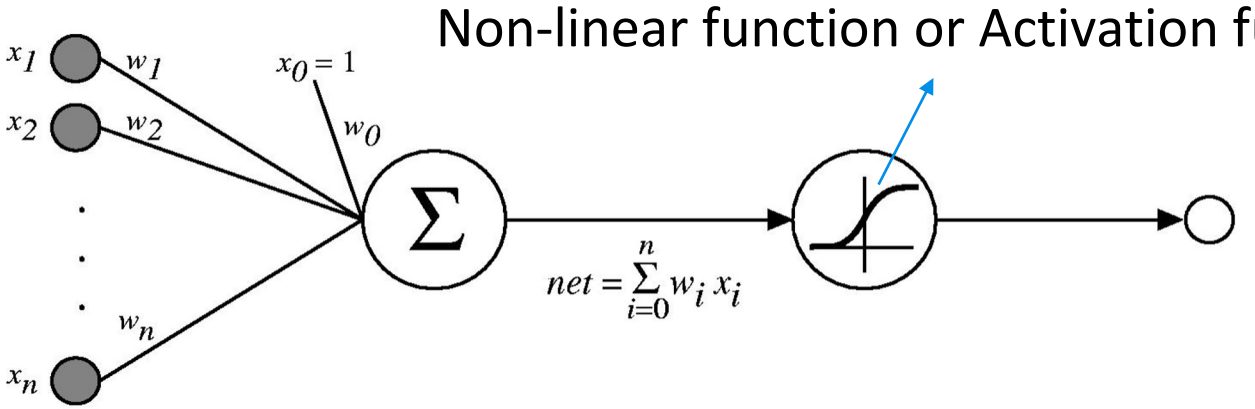


$$o(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + \cdots + w_n x_n > 0 \\ -1 & \text{otherwise.} \end{cases}$$

Sometimes we'll use simpler vector notation:

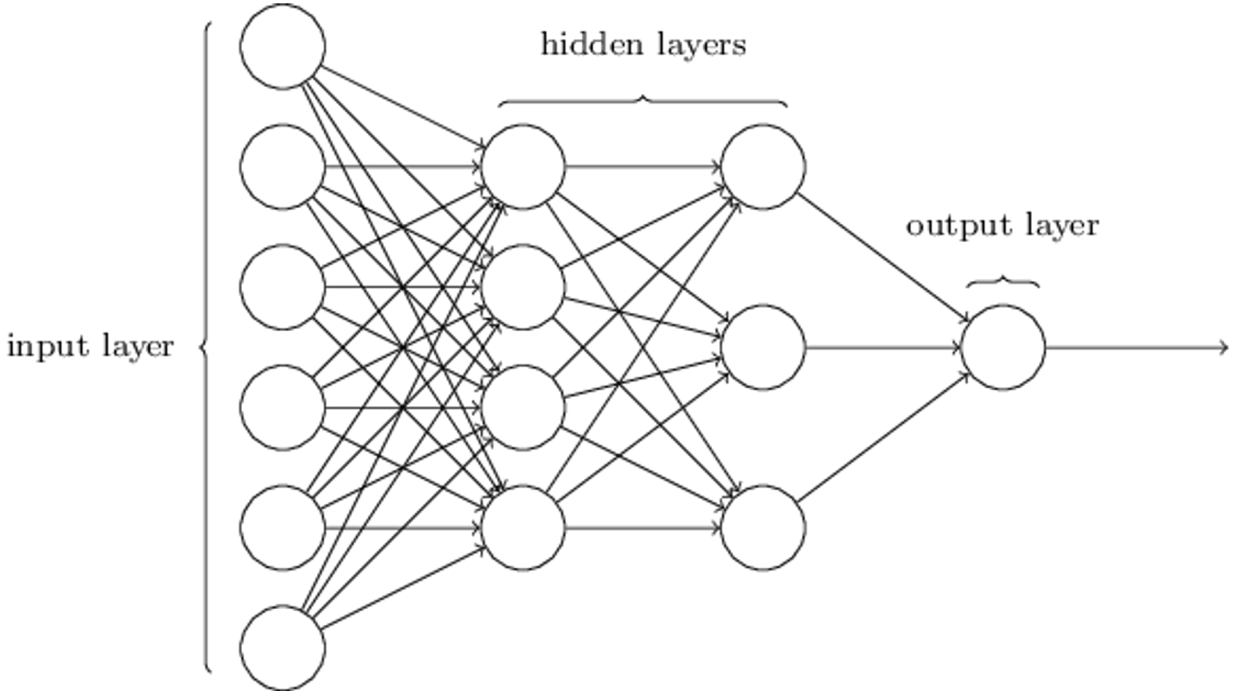
$$o(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \cdot \vec{x} > 0 \\ -1 & \text{otherwise.} \end{cases}$$

Recap: Neural Networks with Activation Functions

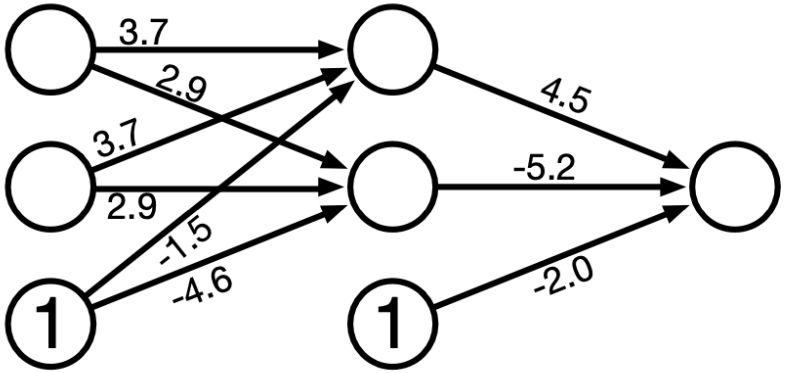


The purpose of the activation function is to introduce ***non-linearity into the network***

Recap: Multi-Layer Perceptron (MLP)



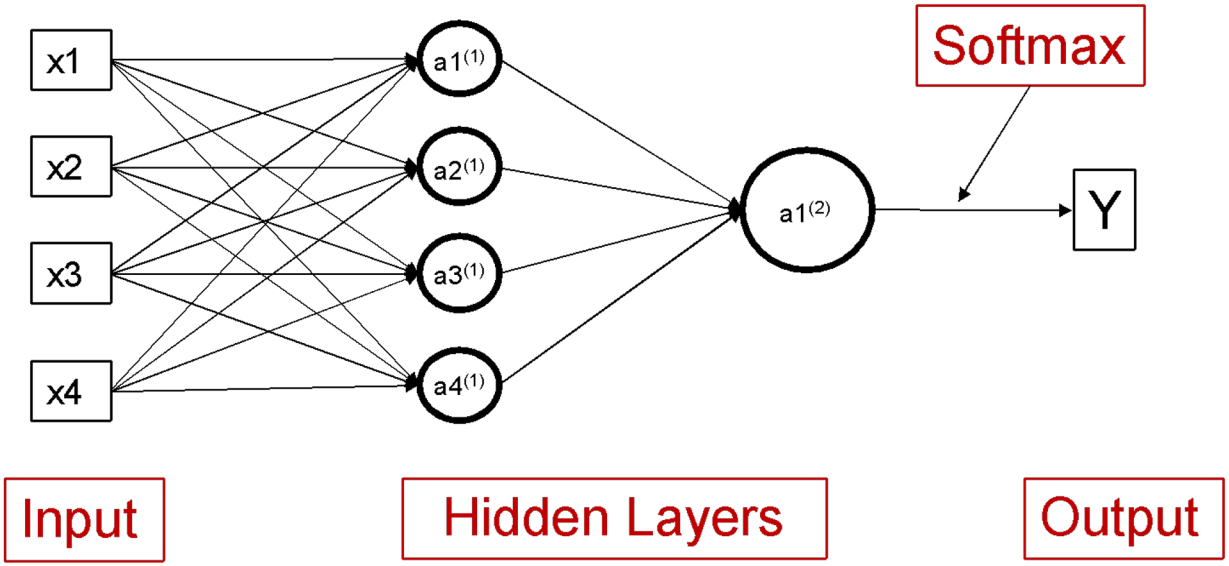
More example



1. Compute the output with sigmoid as the activation function, directly and matrix-based

2. Do on step update with x = (1, 0) y = 1, with binary cross entropy loss, directly and matrix-based

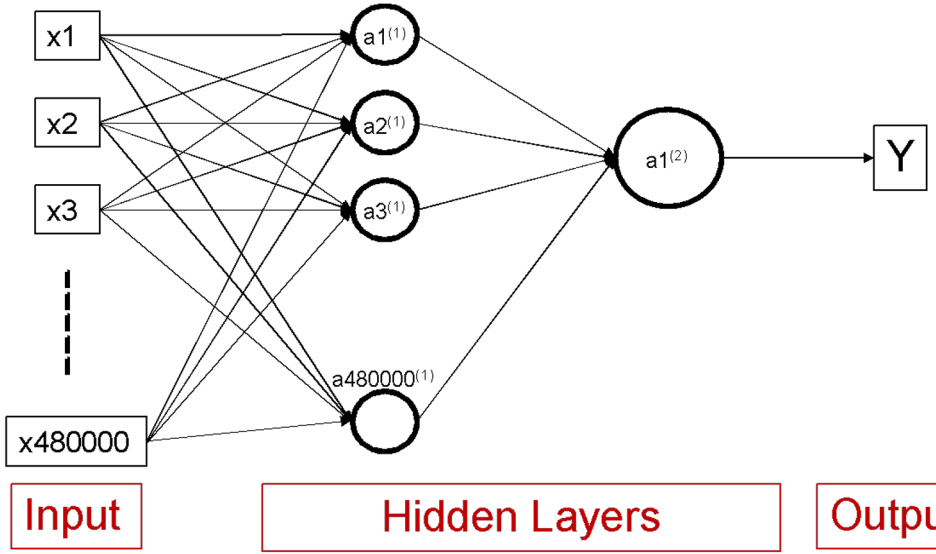
Number of Parameters



$$21 = 4*4 + 4 + 1$$

If the input is an Image?

400 X 400 X 3



Number of Parameters

480000*480000 + 480000 +1 = **approximately 230 Billion !!!**

480000*1000 + 1000 +1 = approximately **480 million !!!**

Convolution Layers

- Filter

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

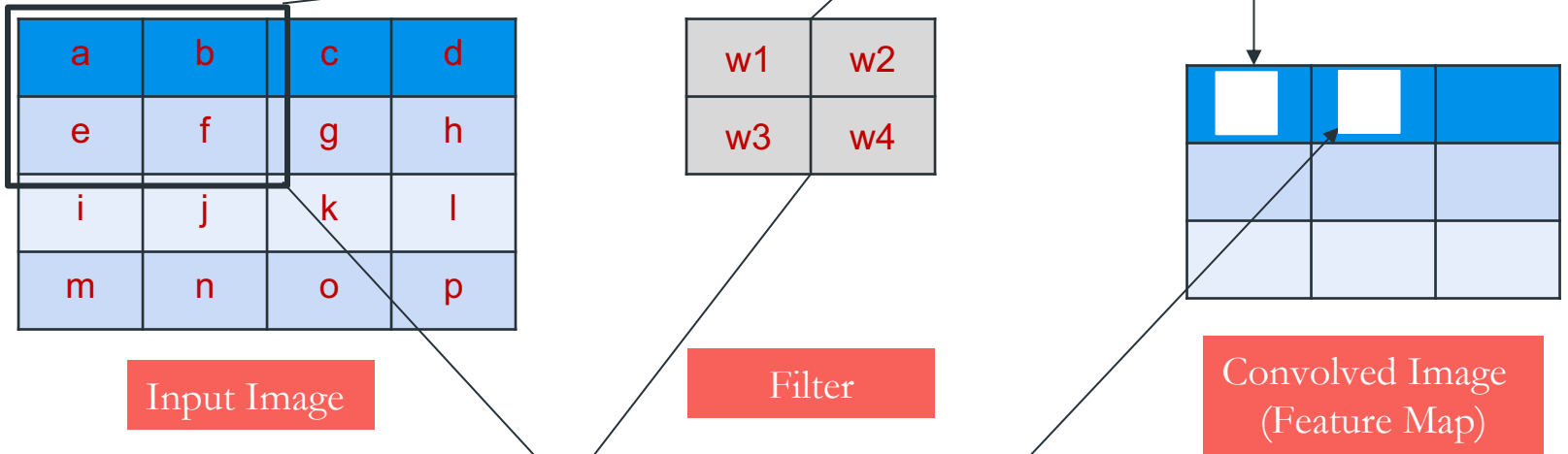
Input Image

- Inspired by the neurophysiological experiments conducted by Hubel and Wiesel 1962.

Convoluted Image

Convolution Layers

$$h1 = f(a * w1 + b * w2 + e * w3 + f * w4)$$

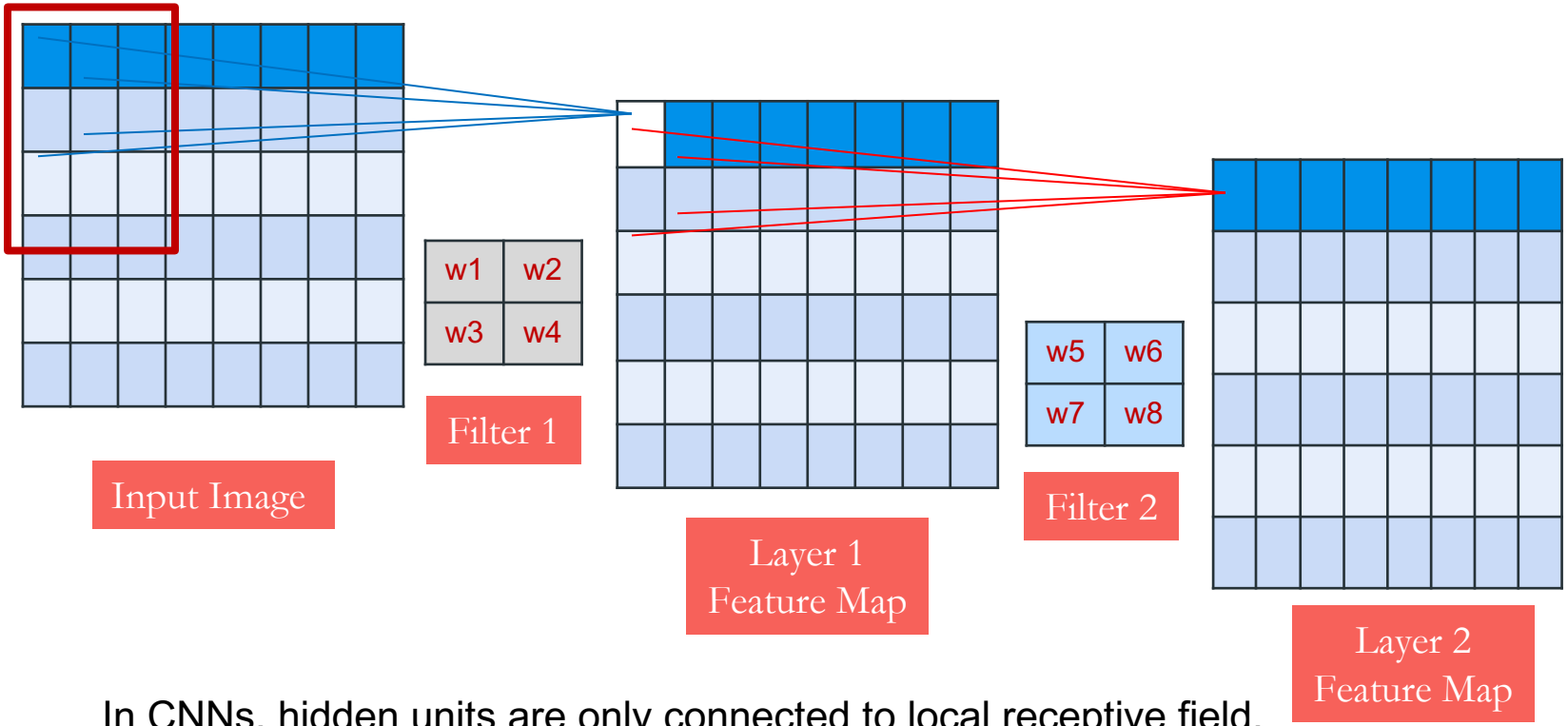


$$h2 = f(b * w1 + c * w2 + f * w3 + g * w4)$$

Number of Parameters for one feature map = 4

Number of Parameters for 100 feature map = 4*100

Convolution Layers



In CNNs, hidden units are only connected to local receptive field.

Pooling Layers

- Max pooling: reports the maximum output within a rectangular neighborhood.
- Average pooling: reports the average output of a rectangular neighborhood.

1	3	5	3
4	2	3	1
3	1	1	3
0	1	0	4

Input Matrix

MaxPool with 2X2 filter with stride of 2

4	5
3	4

Output Matrix

Pooling Layers

- Max pooling: reports the maximum output within a rectangular neighborhood.
- Average pooling: reports the average output of a rectangular neighborhood.

1	3	5	3
4	2	3	1
3	1	1	3
0	1	0	4

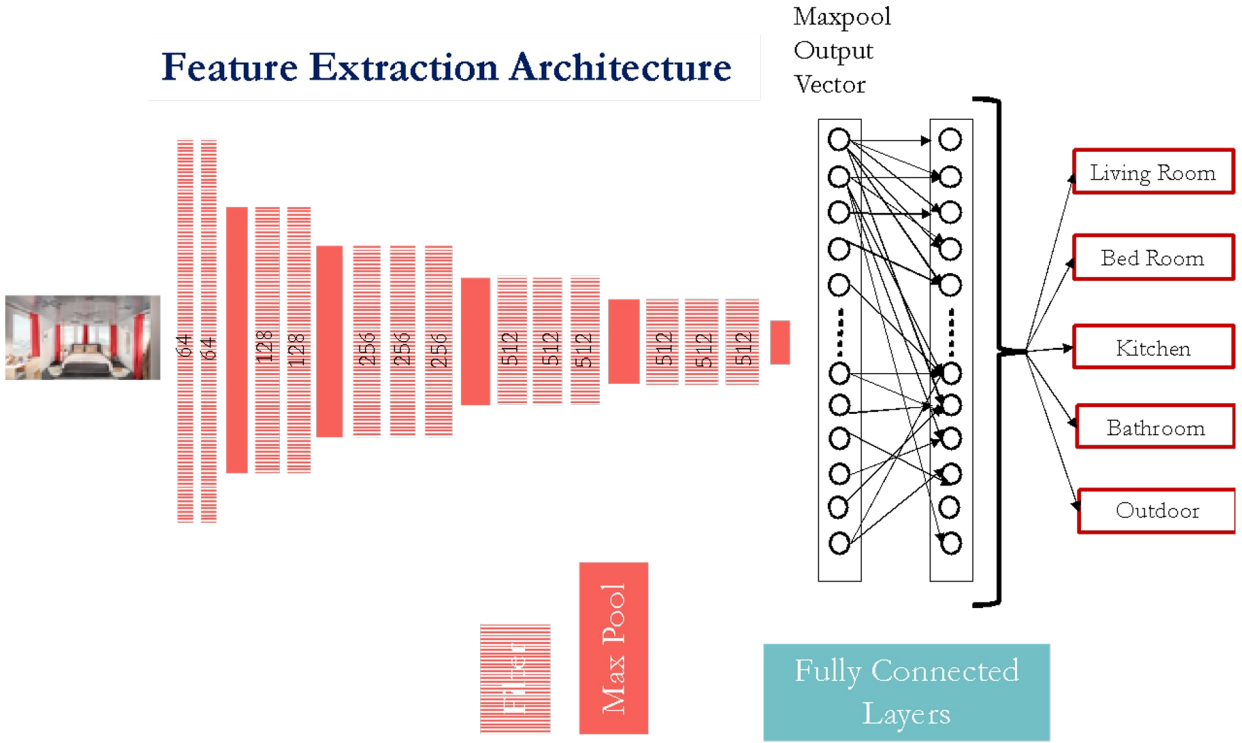
Input Matrix

MaxPool with 2X2 filter with stride of 2

4	5
3	4

Output Matrix

Convolution Neural Networks (1)

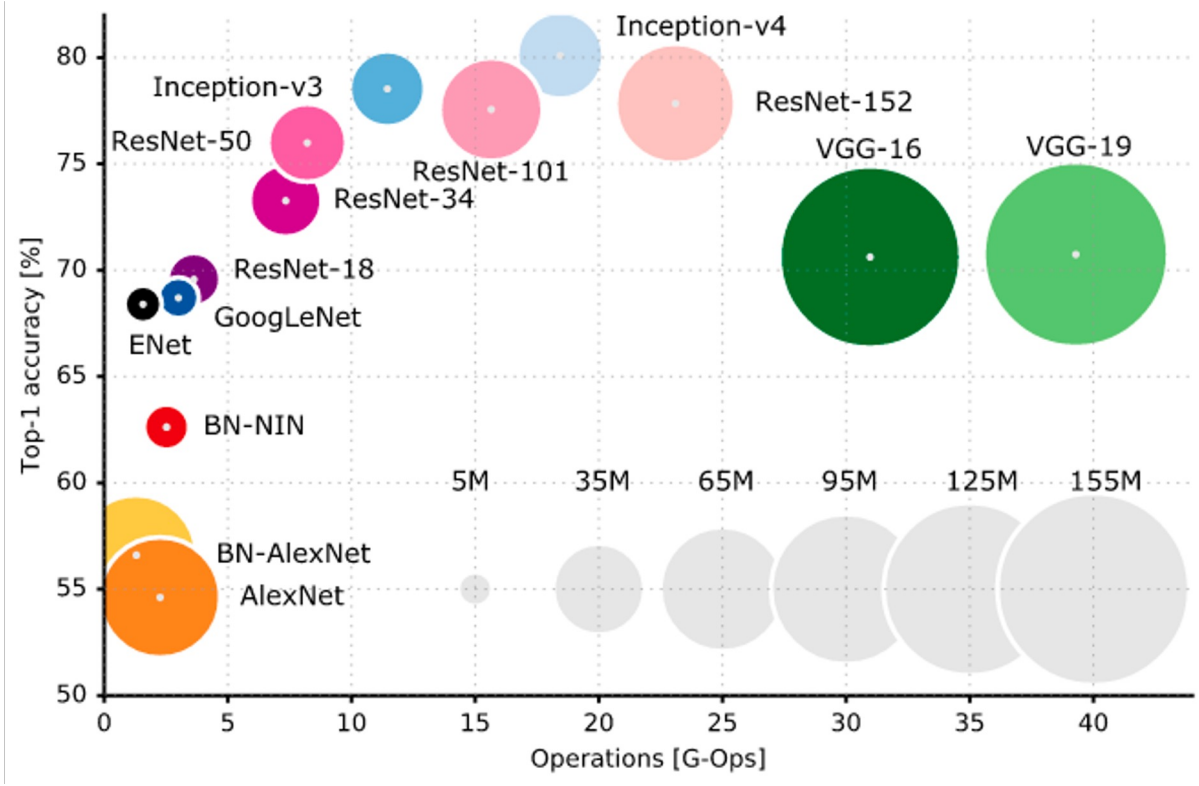


Convolution Neural Networks (2)

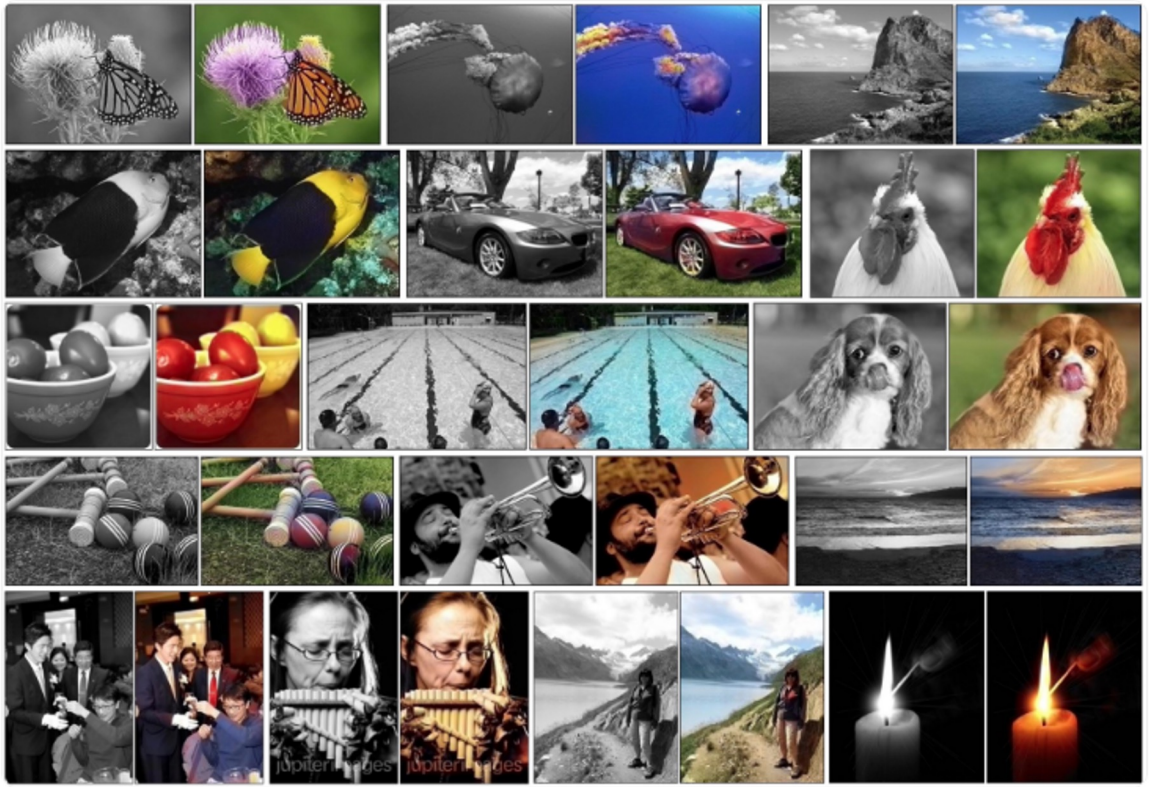
- **Output:** Binary, Multinomial, Continuous, Count
- **Input:** Fixed size, can use padding to make all images same size.
- **Architecture:** Choice is ad hoc
  - requires experimentation.
- **Optimization:** Backward propagation
  - Hyper parameters for very deep model can be estimated properly only if you have billions of images.
    - Use an architecture and trained hyper parameters from other papers (ImageNet or Microsoft/Google APIs etc)
- **Computing Power:** Buy a GPU!!

Read more: https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9

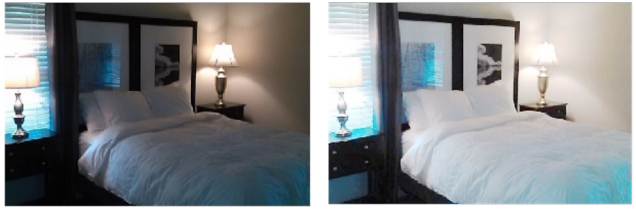
Convolution Neural Networks (3)



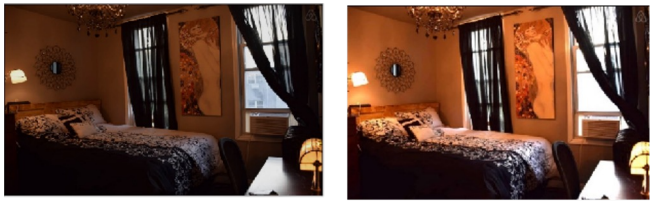
Automatic Colorization of Black and White Images



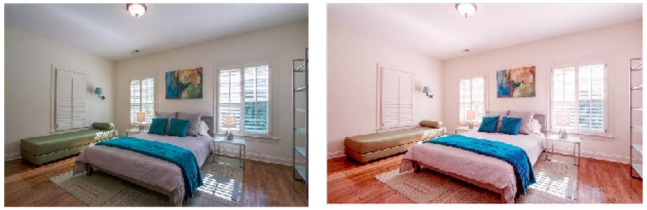
Optimizing Images



Post Processing Feature Optimization
(Color Curves and Details)

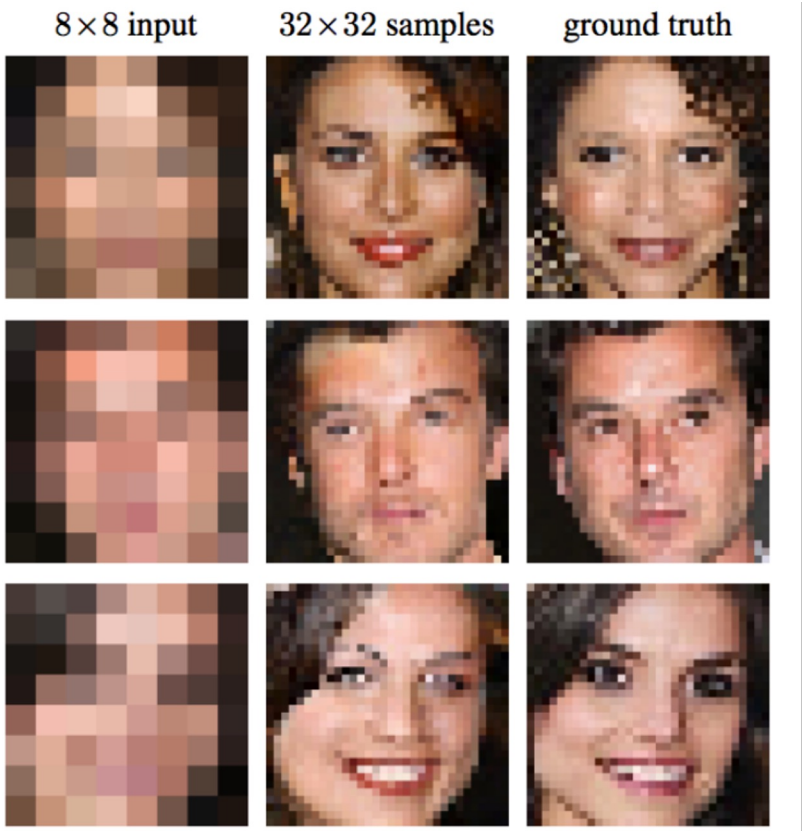


Post Processing Feature Optimization
(Illumination)



Post Processing Feature Optimization
(Color Tone: Warmness)

Optimizing Images



8x8 pixel photos were inputted into a Deep Learning network which tried to guess what the original face looked like. As you can see it was fairly close (the correct answer is under "ground truth").

Limitations of CNN on modeling Sequences

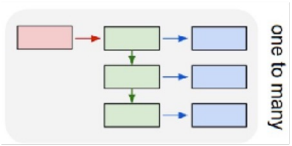
The **limitations** of the **Convolutional Neural Networks**

- Take fixed length vectors as input
- And produce fixed length vectors as output.
- Allow fixed amount of computational steps.

We need to model the data with **temporal or sequential structures** and **varying length of inputs/outputs**

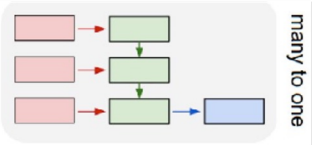
E.g., This movie is very slow in the beginning but picks up pace later on and has some great action sequences and comedy scenes.

Modeling Sequences



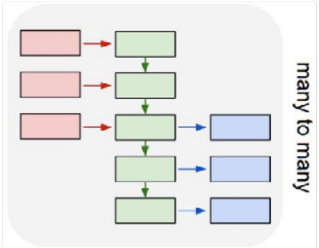
A person riding a motorbike on dirt road

Image Captioning

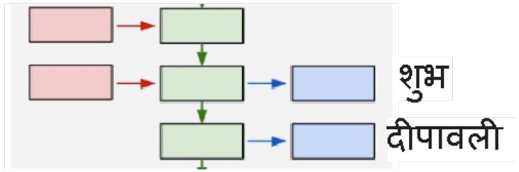


Awesome tutorial. Positive

Sentiment Analysis



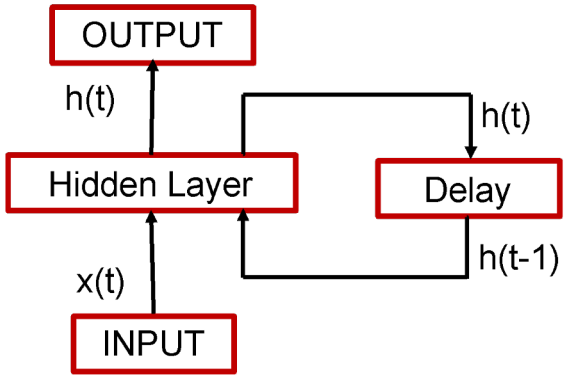
Happy Diwali



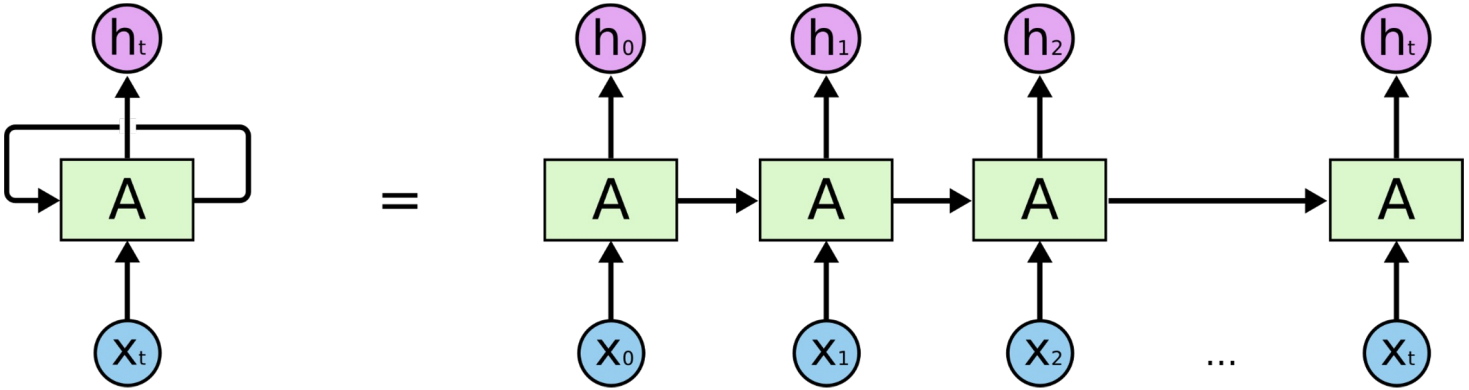
Machine Translation

What is Recurrent Neural Networks?

- Recurrent neural networks are **connectionist models** with the ability to **selectively pass information across sequence steps**, while processing sequential data one element at a time.
- Allows **a memory of the previous inputs** to **persist in the model's internal state** and **influence the outcome**.



RNN - Rolled over time



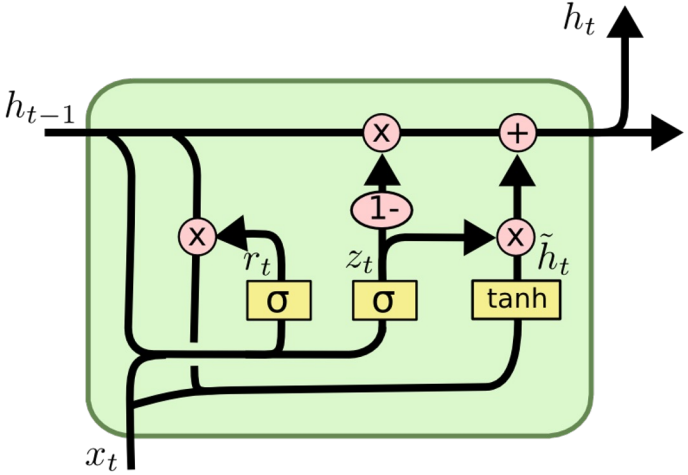
The Vanishing Gradient Problem

- RNN's use back propagation.
- Back propagation uses chain rule.
  - Chain rule multiplies derivatives
- If these derivatives are between 0 and 1 the product vanishes as the chain gets longer.
  - or the product explodes if the derivatives are greater than 1.
- Sigmoid activation function in RNN leads to this problem.
- ReLu, in theory, avoids this problem but not in practice.

Long-Short Term Memory (LSTM)

- LSTM provide solution to **the vanishing/exploding gradient problem**.
- Solution: **Memory Cell**, which is updated at each step in the sequence.
- Three Gates control the flow of information to and from the Memory cell
  - **Input Gate**: protect the current step from irrelevant inputs
  - **Output Gate**: prevents current step from passing irrelevant information to later steps.
  - **Forget Gate**: limits information passed from one cell to the next.

Long-Short Term Memory (LSTM)



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

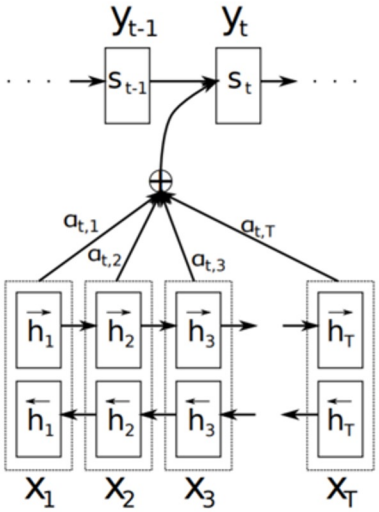
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

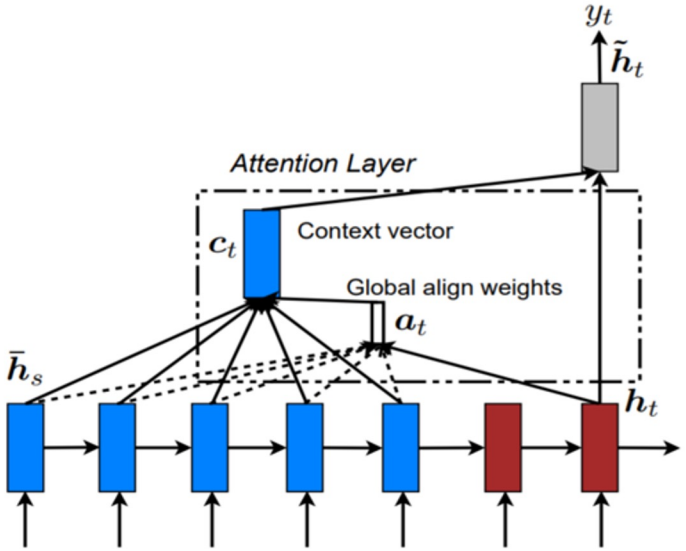
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Reference: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Attention Mechanism



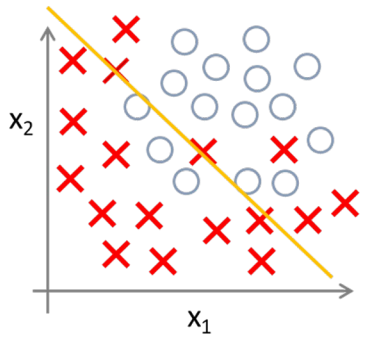
Bahdanau attention mechanism



Luong attention mechanism

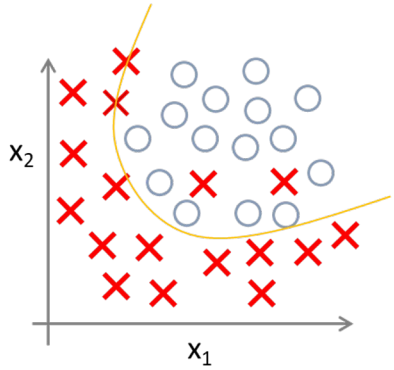
Read more: https://machinelearningmastery.com/the-attention-mechanism-from-scratch/

Underfitting, Overfitting & Goodfit

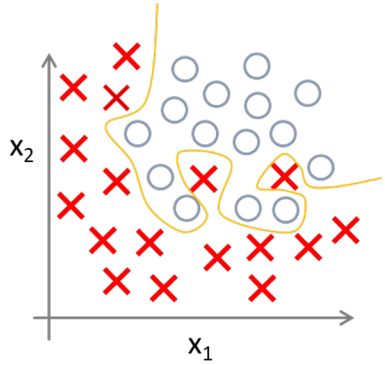


$$h(x) = g(w_0 + w_1 x_1 + w_2 x_2)$$

“Underfitting”



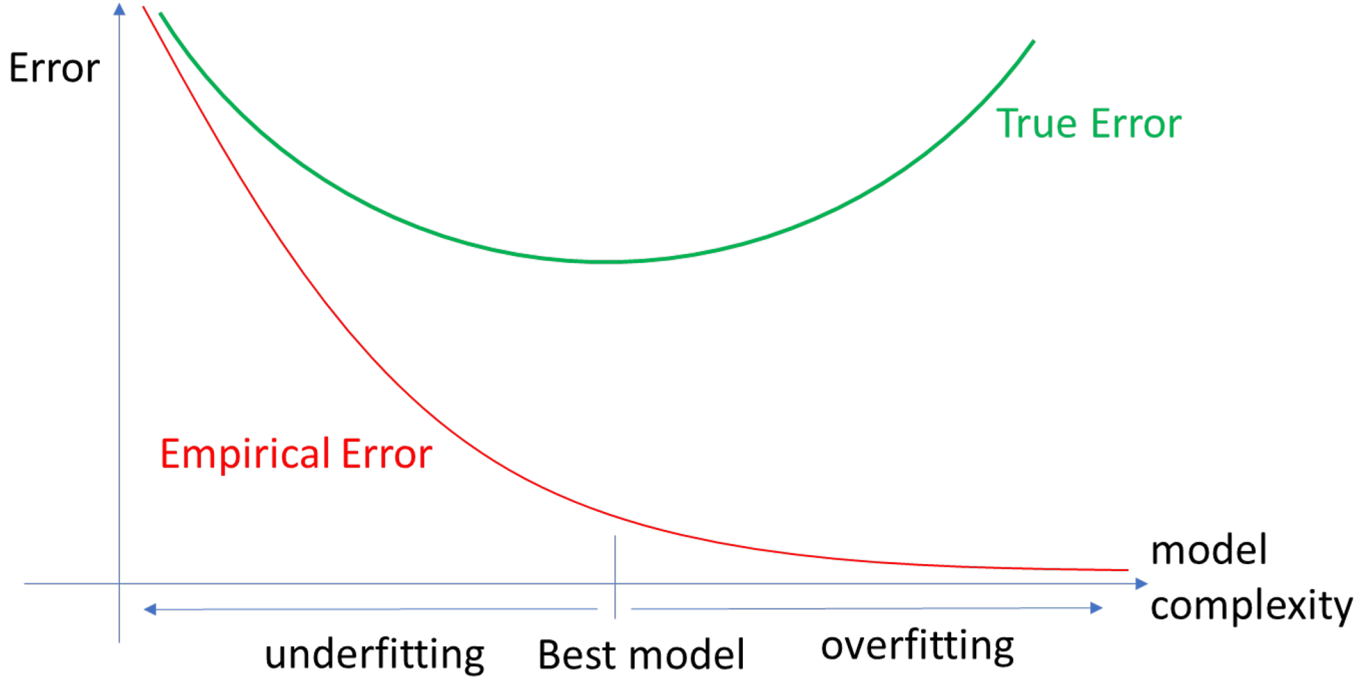
$$g(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + w_5 x1 x_2)$$



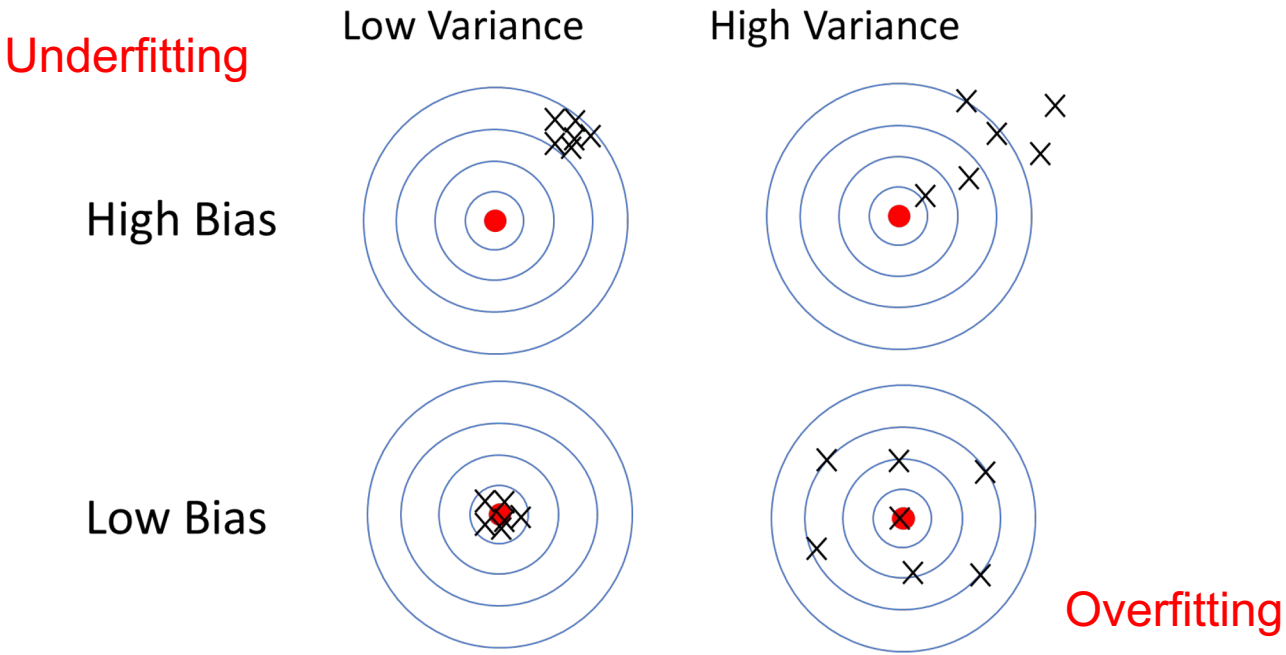
$$g(w_0 + w_1 x_1 + w_2 x_1^2 + w_3 x_1^2 x_2 + w_4 x_1^2 x_2^2 + w_5 x1^2 x_2^3 + w_6 x1^3 x_2 + \ldots)$$

“Overfitting”

The Problem of Model Generalization



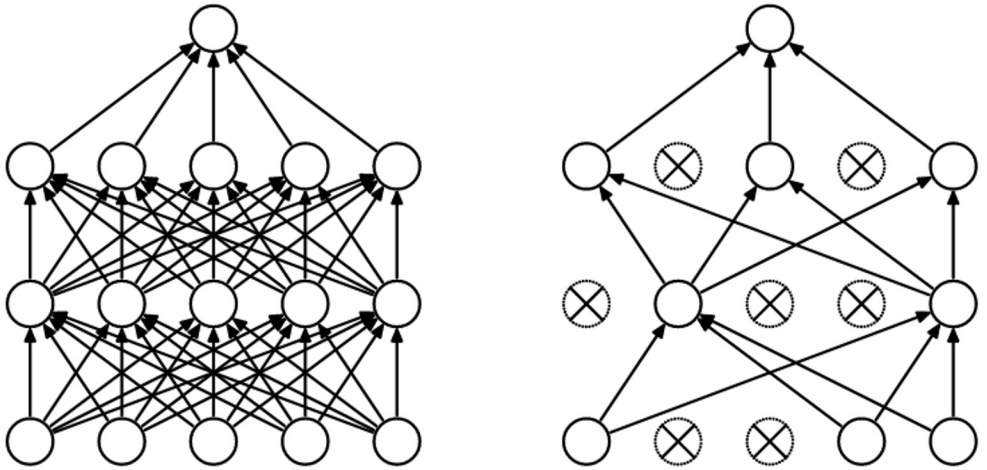
Bias-Variance Trade-Off



Overfitting Reduction for Deep Learning Models

- Dropout
- Batch Nomization
- Early Stopping
- Cross-Validation

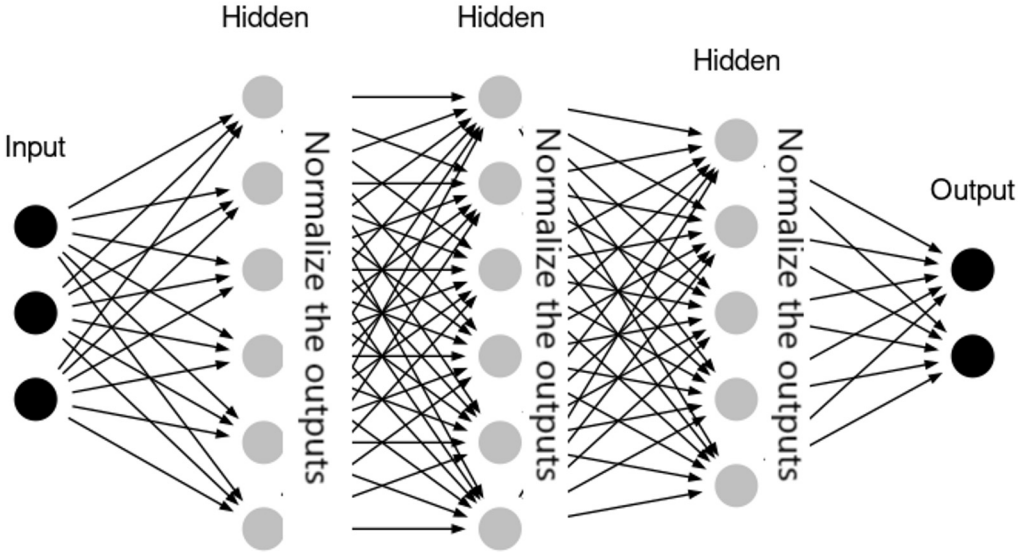
Dropout



(a) Standard Neural Net (b) After applying dropout.

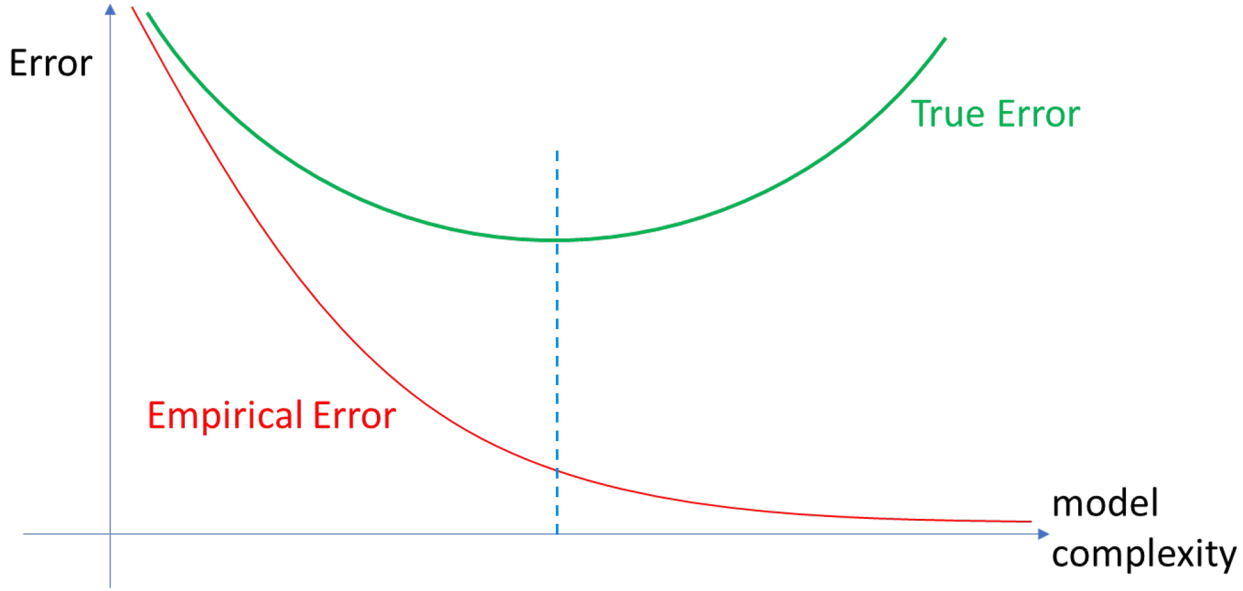
Dropout is a **technique** reduce the model complexity whereby randomly selected neurons are ignored during training. They are “dropped-out” randomly with probability *p*.

Batch Normalization



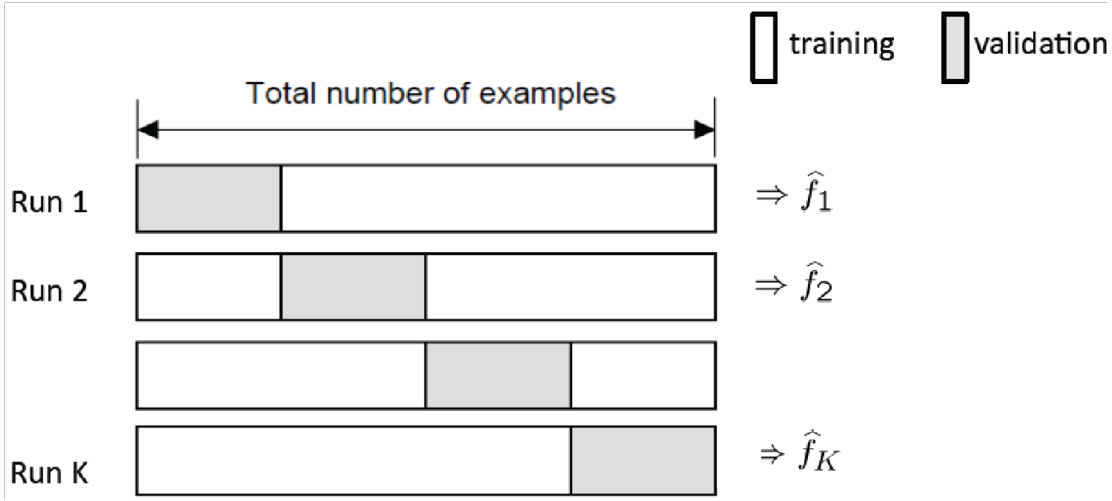
Batch normalization is a **technique** that standardizes the inputs to a layer for each mini-batch. This has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks

Early Stopping



Early stopping is a **technique** where try to stop training as soon as the validation error has stopped reducing

Cross Validation



K-fold cross validation is a **technique** where randomly create K-fold partition of the dataset. Form K hold-out predictors, each time using one partition as validation and rest (K-1) as training datasets

Summary

- Typical Architectures:
  - Convolutional Neural Networks
  - Recurrent Neural Networks
  - Attention Mechanism
- Overfitting Reduction
  - Dropout, Batch Norm
  - Early Stopping
  - Cross Validation

Group Projects: IMDB Genre Classification

- In this project, you will train a model to classify an movie’genre from the movie’s title, its cover image and the user’s input rating. You can look into the README.md for more details. Note that: A movie can have multiple genre
- The dataset is extract from the Movelens 10M dataset, with pre-split train/test files. You should choose to train a model on train set, and test the model on the test set and report the results, using the map@K metrics.
- We will release a baseline in this week as a starting point.

Group Projects: IMDB Genre Classification

Evaluation:
- Report: 40%, including: Preprocess, Approach, Model, Experiments and Results.
- Code: 30%. The code is self-contained. We will train and evaluate your reported results. Therefore, it should be well-documented and reproducbile.
- Video: 30%. Each team should prepared at max-length 10-minute video, which summarizes your report, your main results (i.e, which is the main improvement compared with the baselines, and other available code-base), and how to run and train your code to reproduce the results.

Group Projects: IMDB Genre Classification

Restriction:

- You can not use additional data for training the model. However, you can use pretrained models for specific tasks.
- You can use the available code and others library, however, they should be mentioned clearly and the source should be cited in the report.
- Both the code and the report do not excess 50MB. They would be submitted through the course. The video should be uploaded to the gdrive and included the link in the report.

Group Projects:Timeline

4 weeks:

- 27/11 – 04/12: Register
- 25/12: Handle final report, code, video

Groups: Maximum 3 people

Group Projects:Timeline

Plus points:
- Each team can choose 1 in 3 topics for presentation with +2 points. Max 2 teams/topic. The team register by email, before 04/12 and start to present in the following weeks.

Topic 1: https://paperswithcode.com/sota/multi-label-classification-on-nus-wide
Topic 2: https://paperswithcode.com/sota/multi-label-text-classification-on-aapd
Topic 3: https://paperswithcode.com/sota/multimodal-sentiment-analysis-on-mosi