Deep Learning & Convex Optimization:

A Deep dive into Optimization of DNNs and Analyzing their Capabilities

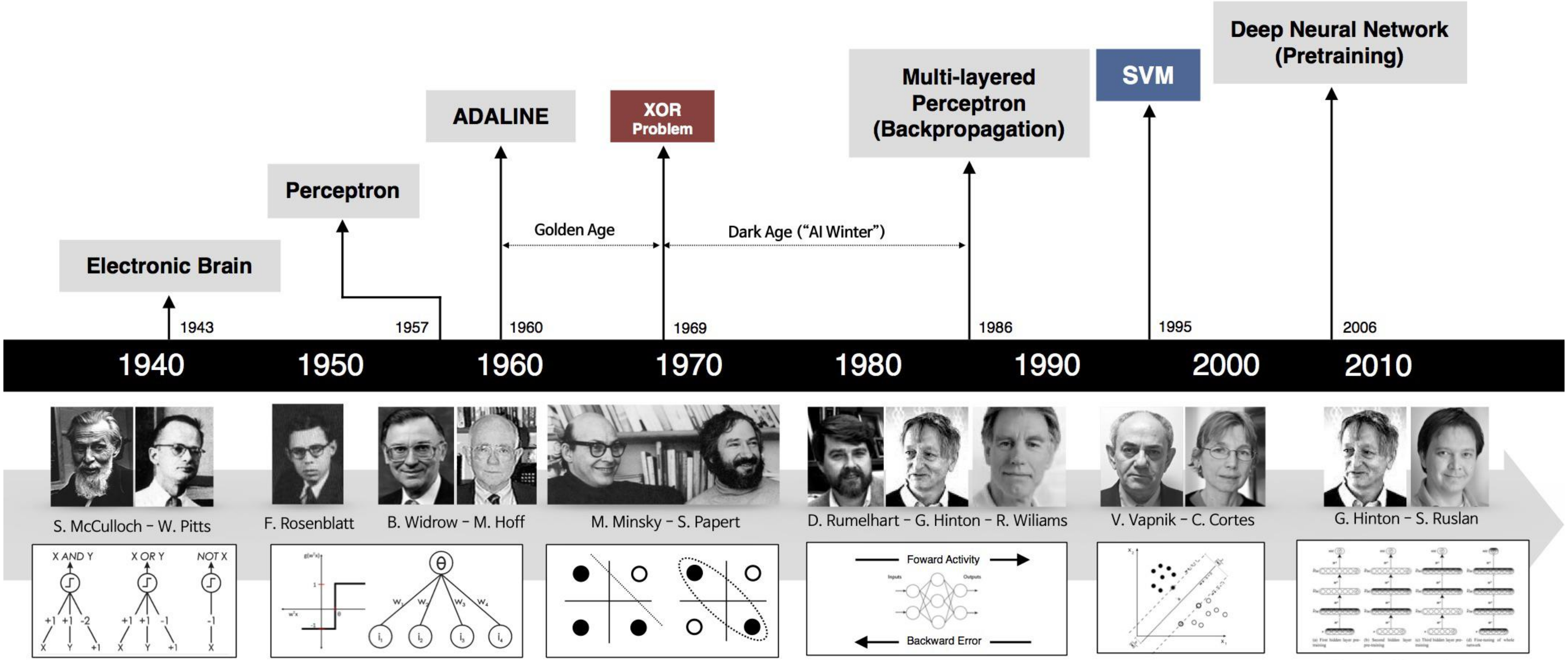
By:

Alireza Gargoori Motlagh - Soroush Atashi

Optimization in Data Science: Final Project

SHARIF UNIVERSITY OF TECHNOLOGY

History of Artificial Neural Networks



Electronic Brain — 1943

Perceptron — 1957

ADALINE — 1960

XOR Problem — 1969

Multi-layered Perceptron (Backpropagation) — 1986

SVM — 1995

Deep Neural Network (Pretraining) — 2006

Golden Age

Dark Age ("AI Winter")

1940 1950 1960 1970 1980 1990 2000 2010

S. McCulloch – W. Pitts

F. Rosenblatt

B. Widrow – M. Hoff

M. Minsky – S. Papert

D. Rumelhart – G. Hinton – R. Wiliams

V. Vapnik – C. Cortes

G. Hinton – S. Ruslan

- Adjustable Weights
- Weights are not Learned

- Learnable Weights and Threshold

- XOR Problem

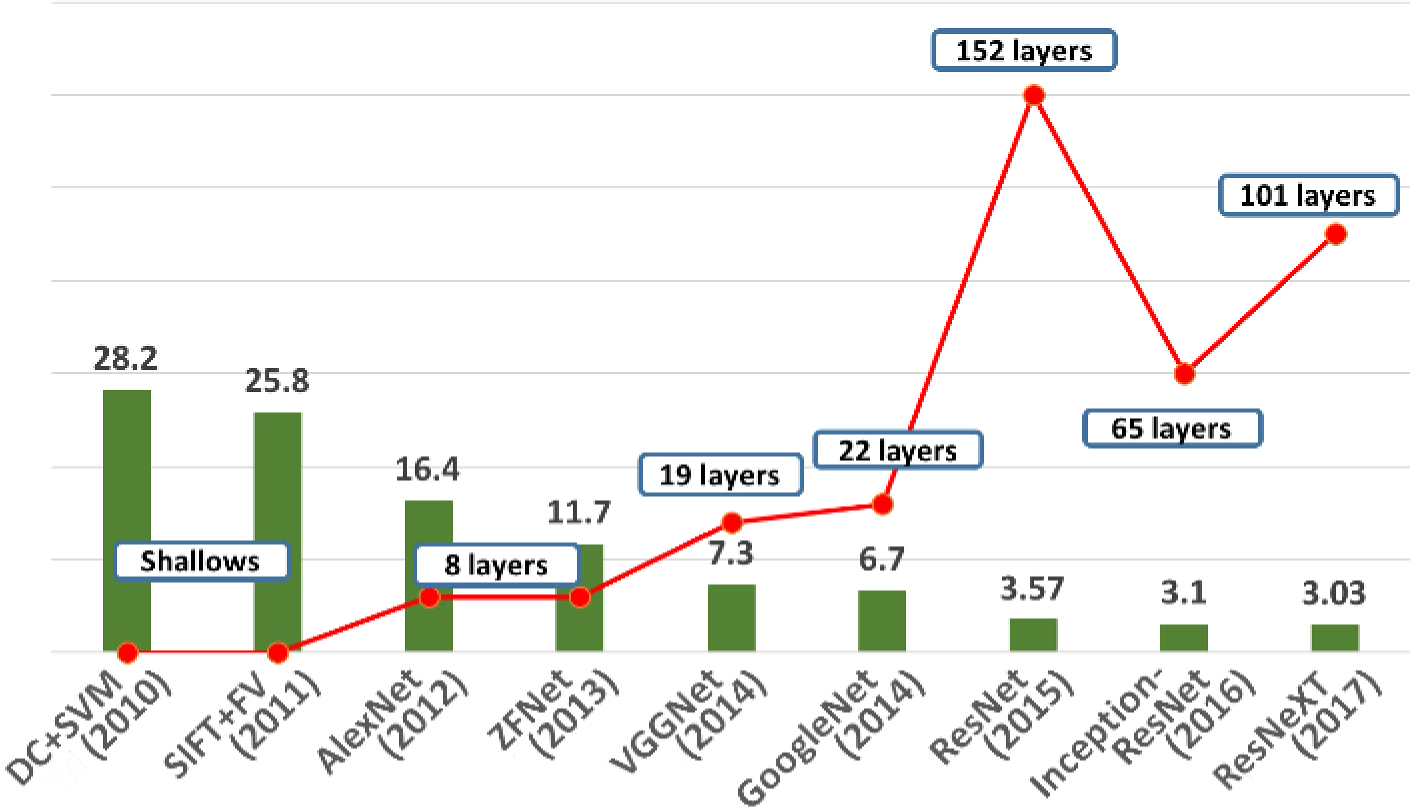
- Solution to nonlinearly separable problems
- Big computation, local optima and overfitting

- Limitations of learning prior knowledge
- Kernel function: Human Intervention

- Hierarchical feature Learning

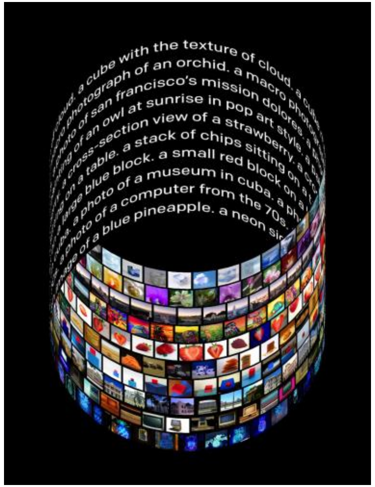
Deep Learning Revolution

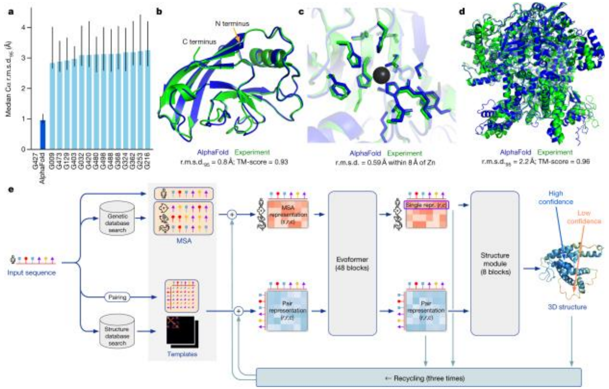
ImageNet Classification ,top-5 error (%)



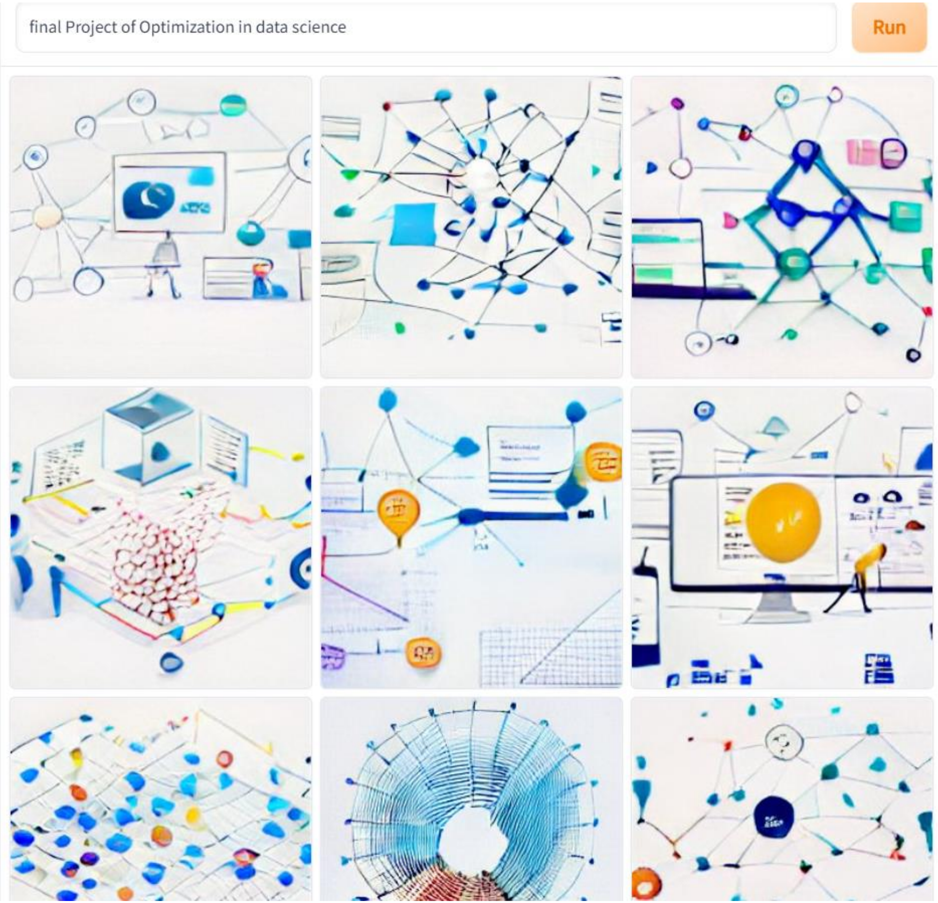
The Impact of Deep Learning

nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO
CONSERVATION
SONGBIRDS À LA CARTE
Illegal harvest of millions of Mediterranean birds
PAGE 452
RESEARCH ETHICS
SAFEGUARD TRANSPARENCY
Don't let openness backfire on individuals
PAGE 459
POPULAR SCIENCE
WHEN GENES GOT 'SELFISH'
Dawkins's calling card 40 years on
PAGE 462
NATURE.COM/NATURE

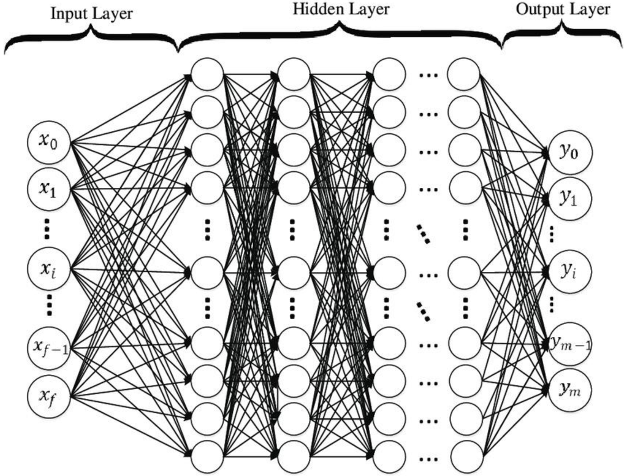

a cube with the texture of cloud. a cu
photograph of an orchid. a macro pho
of san francisco's mission dolores
of an owl at sunrise in pop art style. a
cross-section view of a strawberry. a
table. a stack of chips sitting on a
blue block. a small red block on a
a photo of a museum in cuba. a ph
photo of a computer from the 70s.
photo of a blue pineapple. a neon si


a
Median Cα r.m.s.d.$_{95}$ (Å)
0
1
2
3
4
G427
AlphaFold
G009
G473
G129
G403
G032
G420
G480
G498
G488
G368
G324
G362
G253
G216
b
N terminus
C terminus
AlphaFold Experiment
r.m.s.d.$_{95}$ = 0.8 Å; TM-score = 0.93
c
AlphaFold Experiment
r.m.s.d. = 0.59 Å within 8 Å of Zn
d
AlphaFold Experiment
r.m.s.d.$_{95}$ = 2.2 Å; TM-score = 0.96
e
Input sequence
Genetic database search
MSA
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)

Dall-E: “Final Project of Optimization in Data Science”!

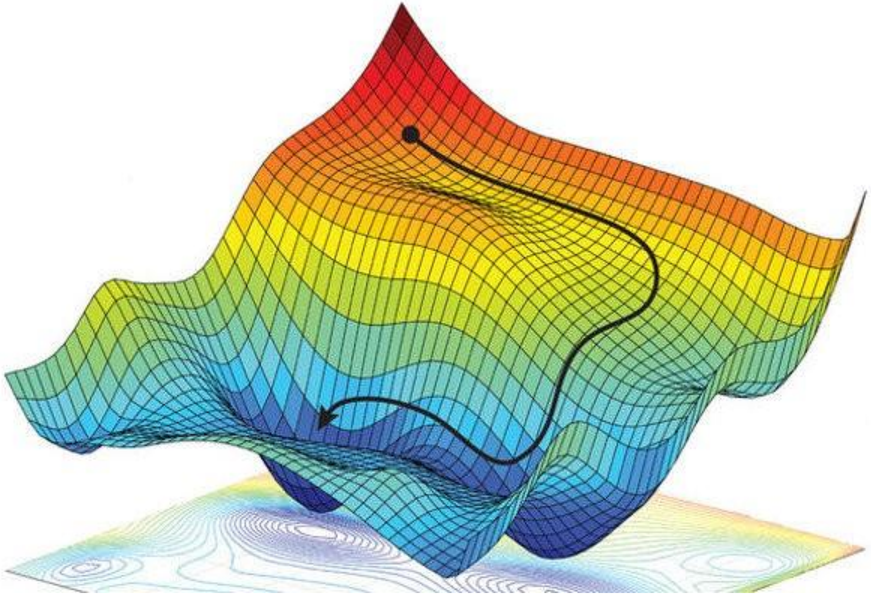


Deep Neural Networks (DNNs) Challenges



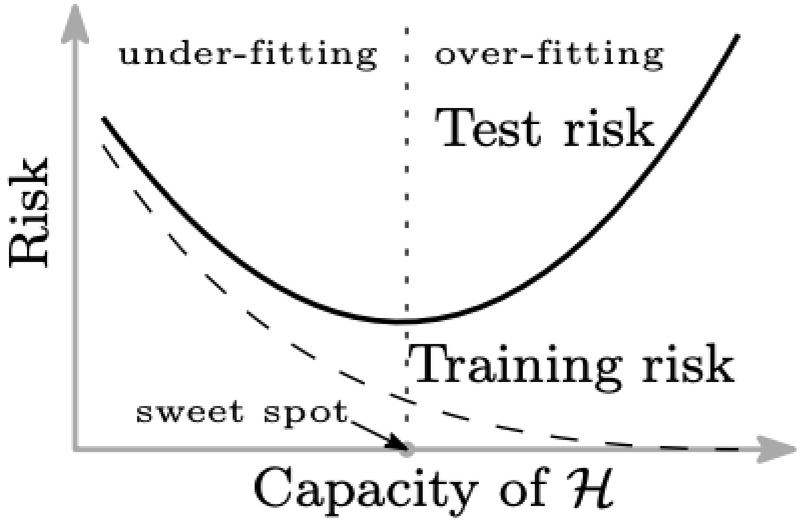
- Highly non-convex loss function
- Extremely high-dimensional problems:
  - 152 layer ResNet-152: 60.2 Million parameters
  - GPT-3 language model: 175 Billion parameters
  - BAAI multi-modal model: 1.75 Trillion parameters
- Challenging to train
  - GPT-3 is estimated to cost $12 Million for a single training run

Non-Convexity of DNNs Cost Function



However, DNNs are the most powerful and state-of-the-art algorithms which are widely used. How such models with Millions/Billions/Trillions of parameters and many critical points are not usually overfitted?

Conventional Machine Learning Theory: Bias-Variance Tradeoff



- **Generalization :** How well a trained model perform on unseen test data?

Conventional ML theory states that the gap between generalization error and empirical error is often bounded by $O(\sqrt{|\mathcal{H}|}/m)$

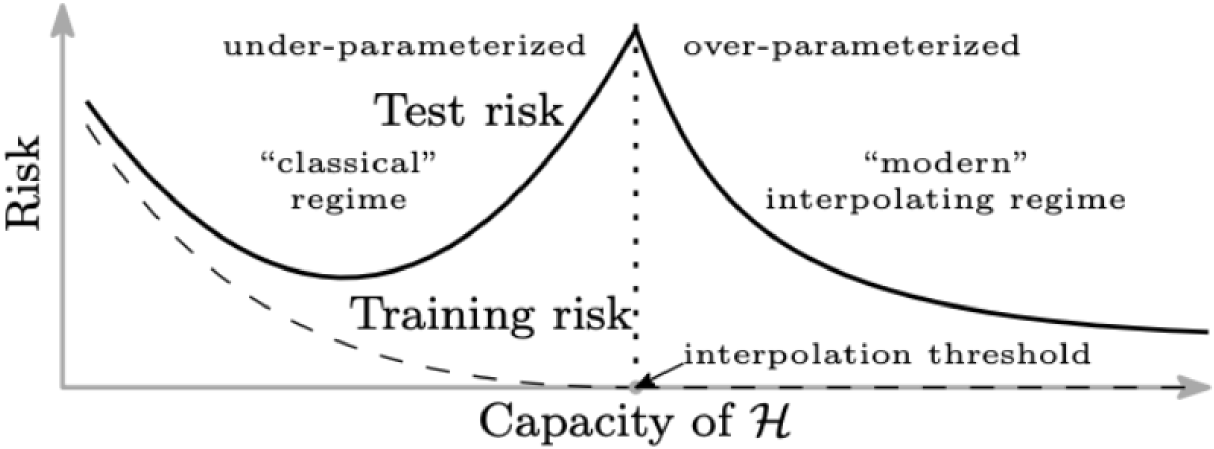
- m: Number of i.i.d. training samples $|\mathcal{H}|$: Complexity of hypothesis space

Contradiction for DNNs?

Over-Parameterization

- **Over-Parameterization**: the regime where the number of model parameters is greater than the number of training examples.

Double Descent Curve



We'll discuss two possible reasons behind this phenomenon in details; but before that, let's validate this curve in some special setting.

Double Descent Curve: A new curve for model assessments?

In this section we discuss the double descent risk curve in the context of neural networks and implement the model of the paper to verify this phenomenon.

- **Random Fourier Features (RFF)**

can be viewed as a class of two-layer neural networks with fixed weights in the first layer. $(h\colon\ \mathbb{R}^d \to \mathbb{C})$

$$h(x) = \sum\nolimits_{k=1}^{N} a_k \phi(x; v_k)$$

$v_1, \ldots, v_N$ are sampled independently from standard normal distribution.

$$\mathbf{\Phi} = \begin{bmatrix} \cos(v_1^\top x_1) & \sin(v_1^\top x_1) & cos(v_2^\top x_1) & \sin(v_2^\top x_1) & \cdots & \cos(v_N^\top x_1) & \sin(v_N^\top x_1) \\ \cos(v_1^\top x_2) & \sin(v_1^\top x_2) & cos(v_2^\top x_2) & \sin(v_2^\top x_2) & \cdots & \cos(v_N^\top x_2) & \sin(v_N^\top x_2) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \cos(v_1^\top x_n) & \sin(v_1^\top x_n) & cos(v_2^\top x_n) & \sin(v_2^\top x_n) & \cdots & \cos(v_N^\top x_n) & \sin(v_N^\top x_n) \end{bmatrix}$$

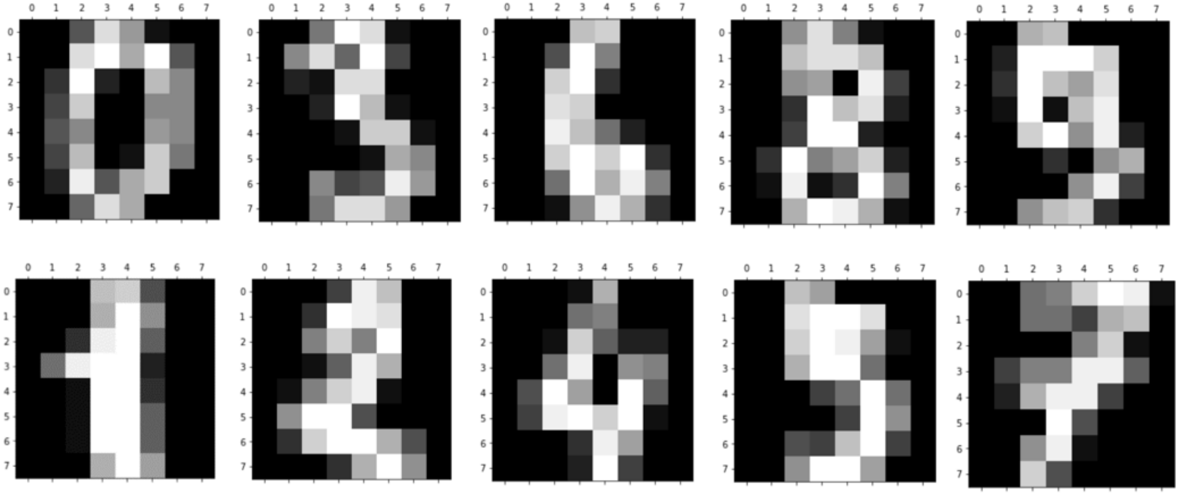
$$\mathbf{\Phi} \in \mathbb{R}^{n \times 2N}$$

$$h(x) = \ a^T \mathbf{\Phi}(x)$$

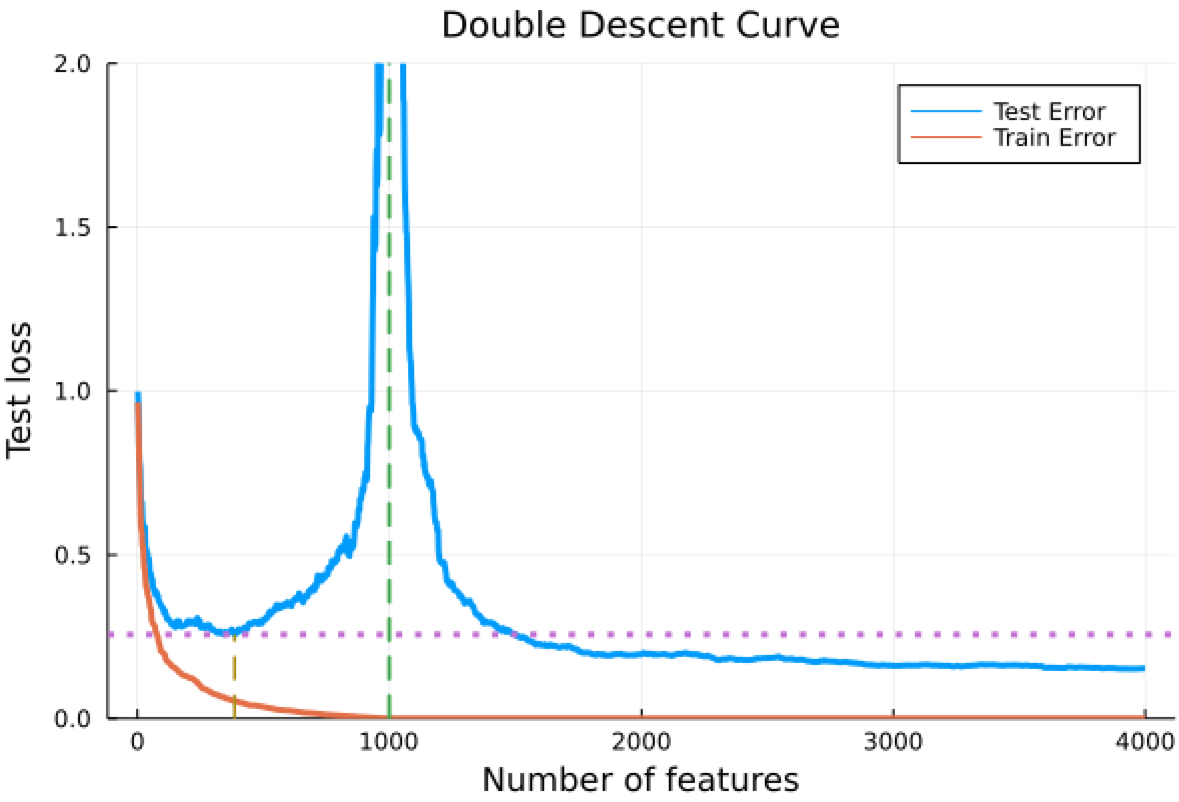
Implementation of Double Descent Curve

"Our learning procedure using $\mathcal{H}_N$ is as follows. Given data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$ from $\mathbb{R}^d \times \mathbb{R}$, we find the predictor $h_{n,N} \in H_N$ via ERM with squared loss. That is, we minimize the empirical risk objective $\sum_{i=1}^{N}(h(x_i) - y_i)^2$ over all functions $h \in \mathcal{H}_N$. When the minimizer is not unique (as is always the case when 2N > n), we choose the minimizer whose coefficients $(a_1, a_2, \ldots, a_N)$ have the minimum l_2 norm."

- Dataset: MNIST
- Objective: Recognize curvy digits from those with angles



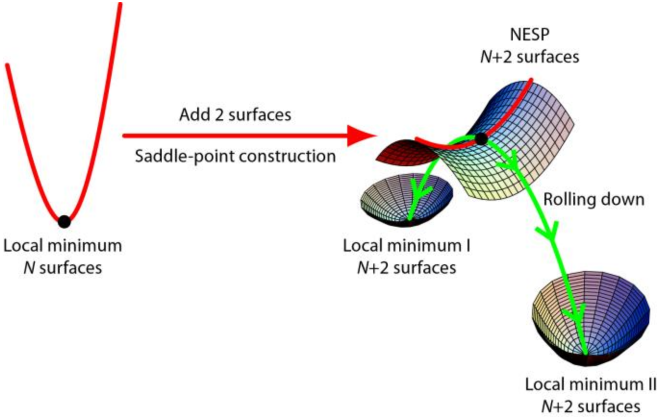
Results of Double Descent Curve



Why Over-Parameterization is an opportunity?

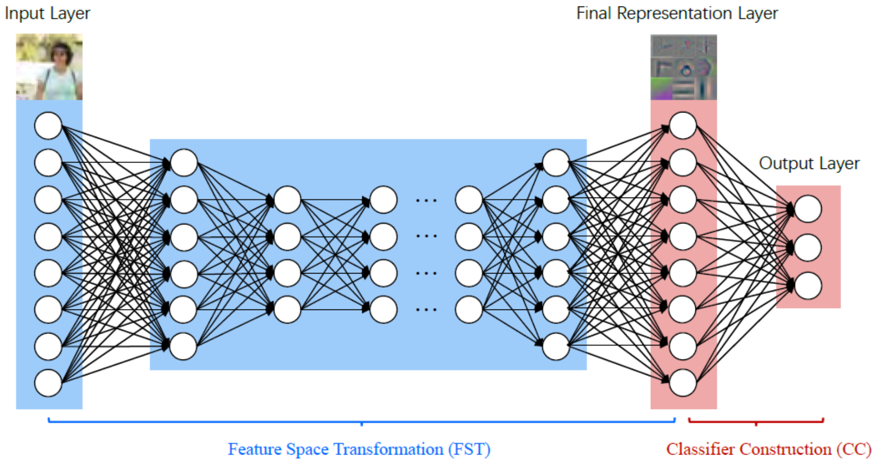
There is no certain reason behind this phenomenon and the great success of deep learning models in spite of their highly redundant parameters is still a mystery; but we can argue and propose 2 possible reasons:

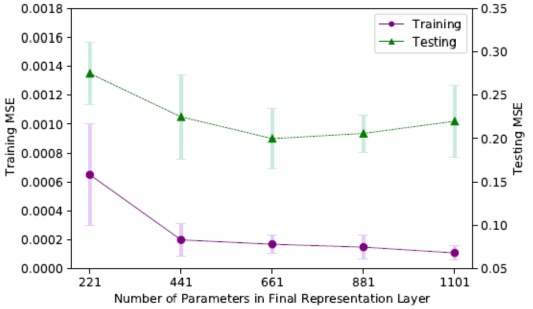
1. **Initializations and Saddle Points**
   - damage coming from some poorly initialized or optimized parameters can be undone by the proper optimization of others.
   - Conversion of possible Local Minimums to Saddle Points and use of SGD to escape saddle points

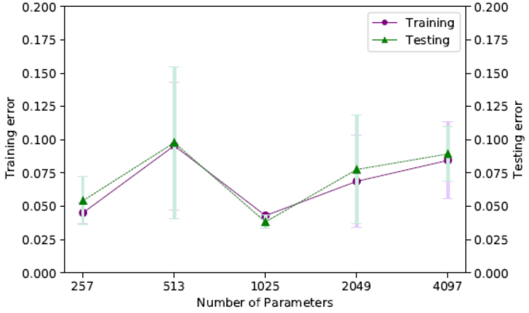


Why Over-Parameterization is an opportunity?

2. Representation Learning


Input Layer
Final Representation Layer
Output Layer
...
Feature Space Transformation (FST)
Classifier Construction (CC)





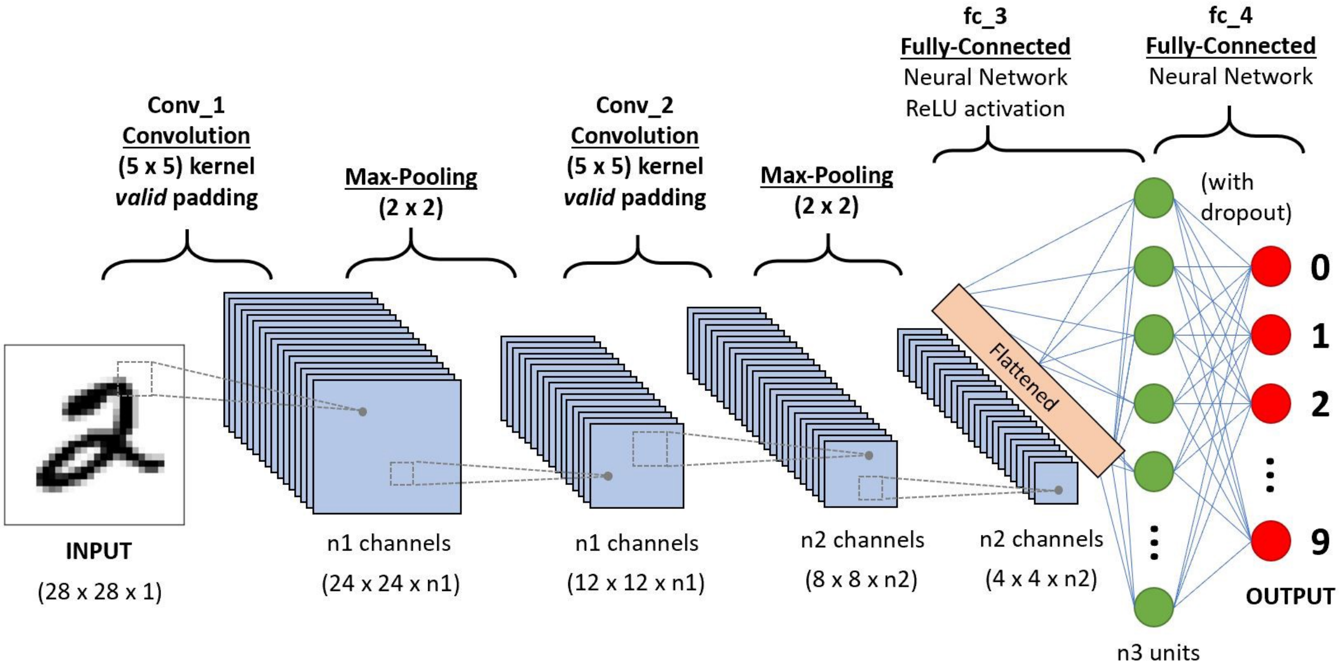
Convex Optimization and Neural Networks

NeurIPS 2019: out of all the accepted papers, **32** papers are related to convex optimization.

- **Convex Optimization and Shallow Neural Networks:**
  - Consist of 2-3 layers in total (not deep!)
  - Universal Approximation Theory: Shallow neural networks are universal approximators.
  - Computational Complexity
  - No feature space transformation due to the lack of number of hidden layers

However, their results could be generalized to multiple hidden layer and DNNs (with some modifications) as well in the near future.

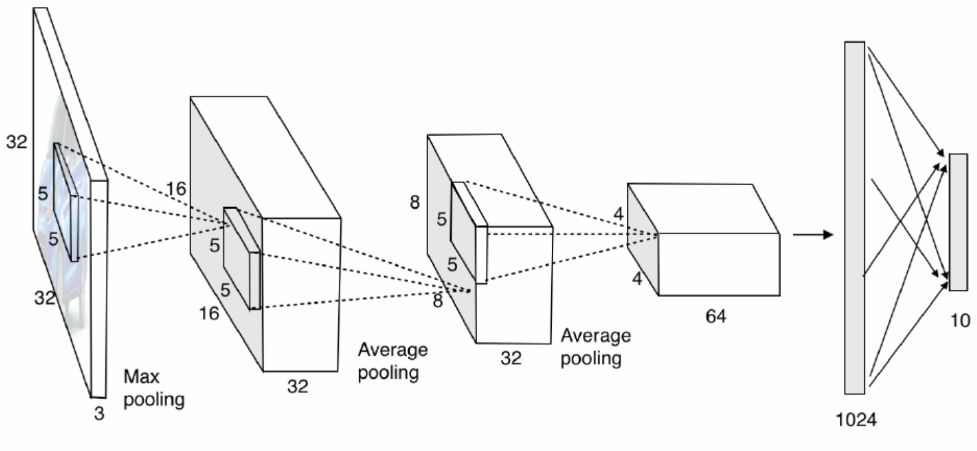
CNNs: What are convolutional neural networks?



CNNs and Convex Optimization: Enhancing the performance of deep learning models using convex optimization

- We can greatly enhance the performance of deep learning models using convex optimization although these models are highly non convex

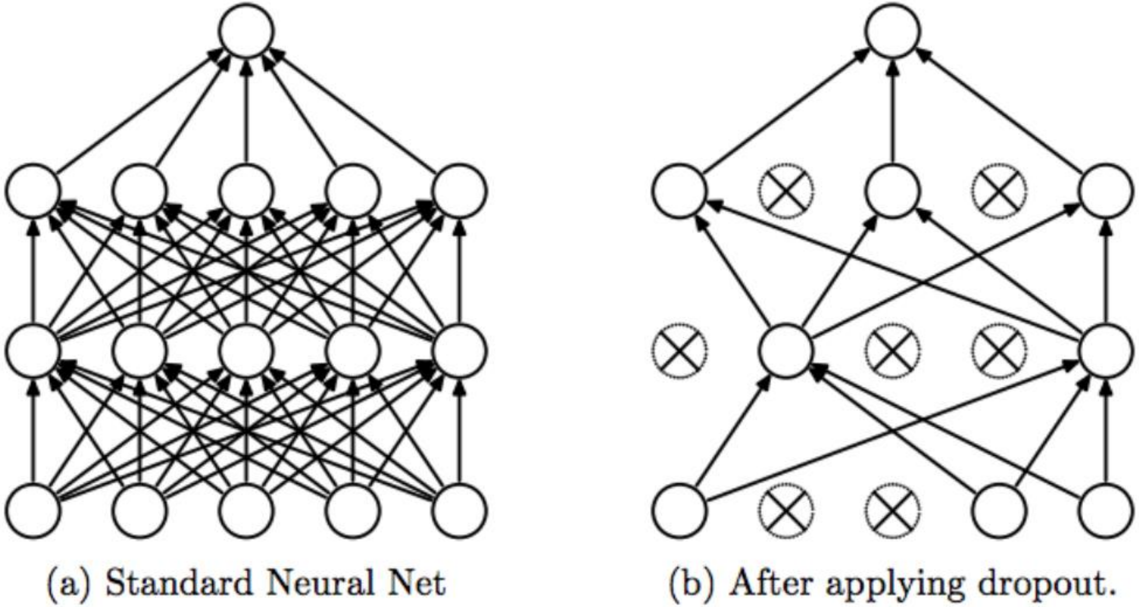
- We can combine dropout and convex optimization in order to achieve better results



What is dropout?

Dropout is a regularization method that approximates training a large number of neural networks with different architectures in parallel.

- We randomly ignore some neurons in the training process to avoid overfitting.



Combining Dropout and Convex Optimization

- By using the dropout model, we achieve sub-models.

Now we can either combine them with the same weight, or use convex optimization. (each column of P indicating the predictive probabilities of each sub-model)

$$\begin{aligned} \min_{l} \quad & \textstyle\sum_{i=1}^{N} \|P_i \cdot l - y_i\|_2^2 \\ \text{s.t.} \quad & l \;\geq\; 0 \end{aligned}$$

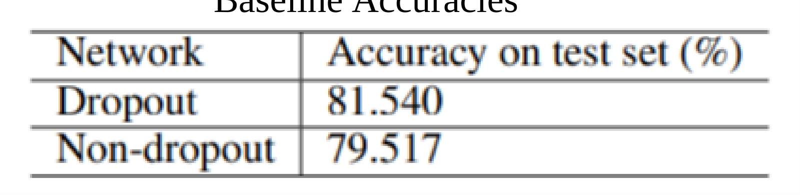
- Quadratic Program, hence can be solved efficiently with convex solvers.

$$\begin{aligned} \min_{l} \quad & l^t P^t P l - 2y^t P l + y^t y \\ \text{s.t.} \quad & l \;\geq\; 0 \end{aligned}$$

Results: How do the introduced methods affect the accuracy of our model?

- For all the experiments, the training and test data are CIFAR-10 dataset. There are 50,000 images in training set and 10,000 images in test set.

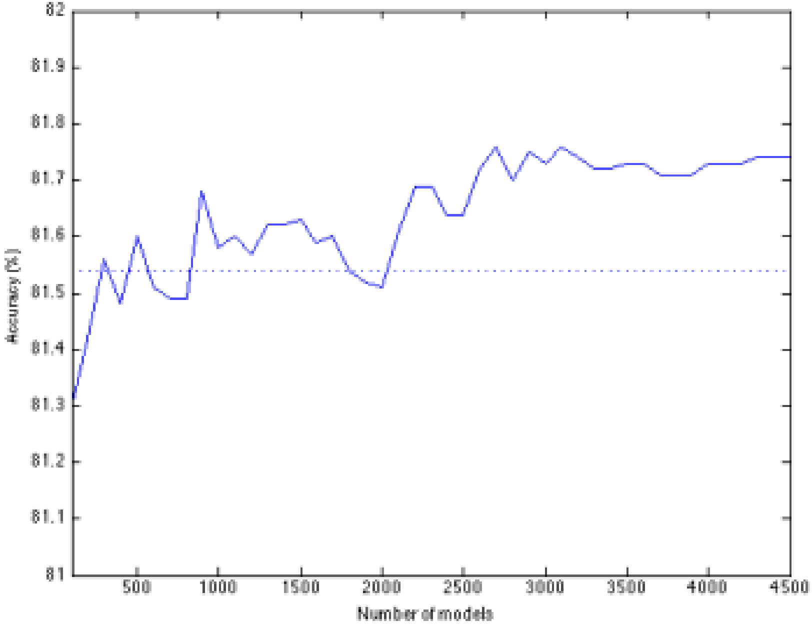
Baseline Accuracies



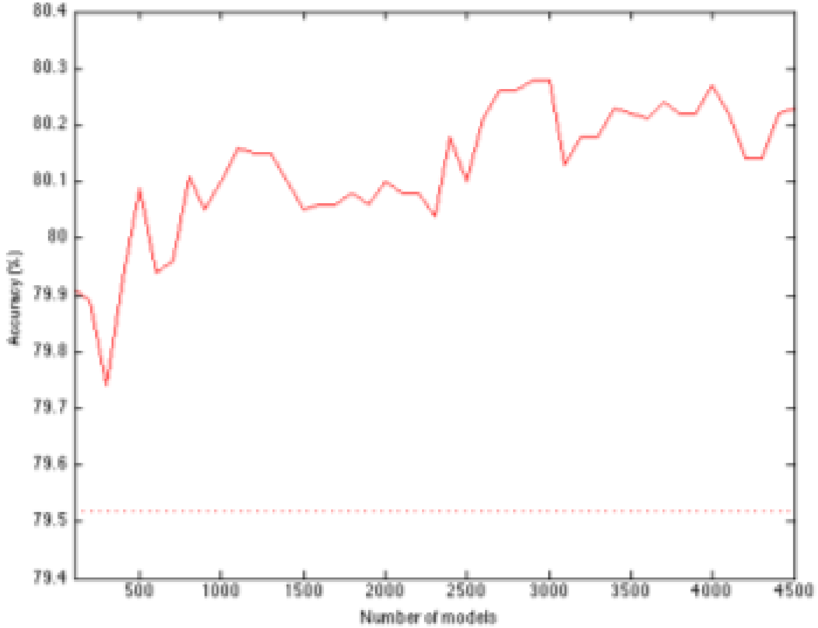
Network	Accuracy on test set (%)
Dropout	81.540
Non-dropout	79.517

- By combining the sub-models, we can increase the accuracy in the network with dropout by 0.22% and in the network without drop out by 0.76%.

Results: Accuracy vs Number of Sub-models



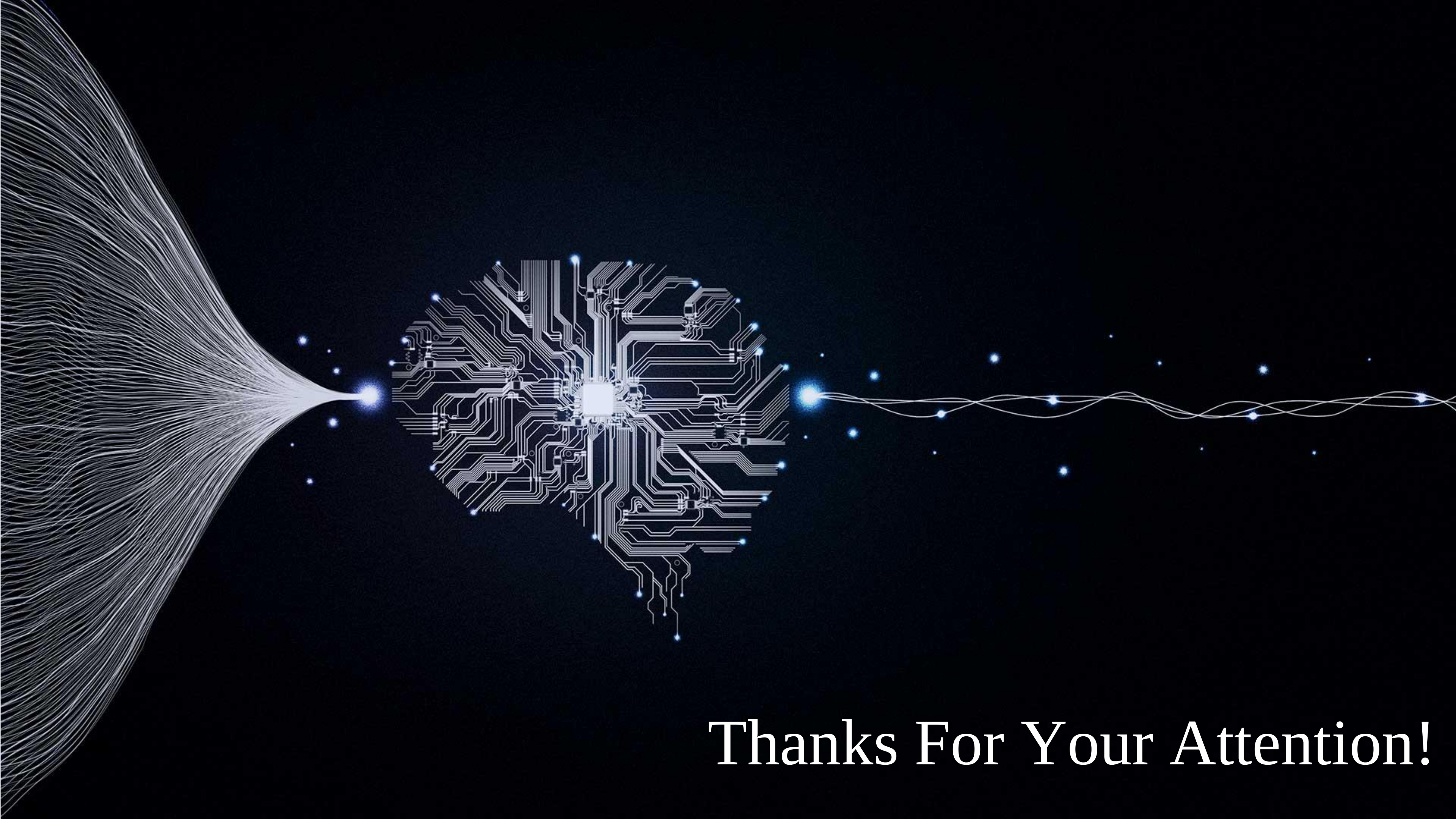
(a) Accuracy-number of sub-models for dropout network



(b) Accuracy v.s. number of sub-models for nondropout network

Conclusion

- We reviewed a brief history of deep learning and its important role
- We talked about over-parametrization and its use in deep learning models
- We introduced CNNs and how convex optimization can improve their performance
- How convex optimization can further help improve deep learning models still needs to be studied massively in order to use convex optimization methods is DNNs as well



Thanks For Your Attention!