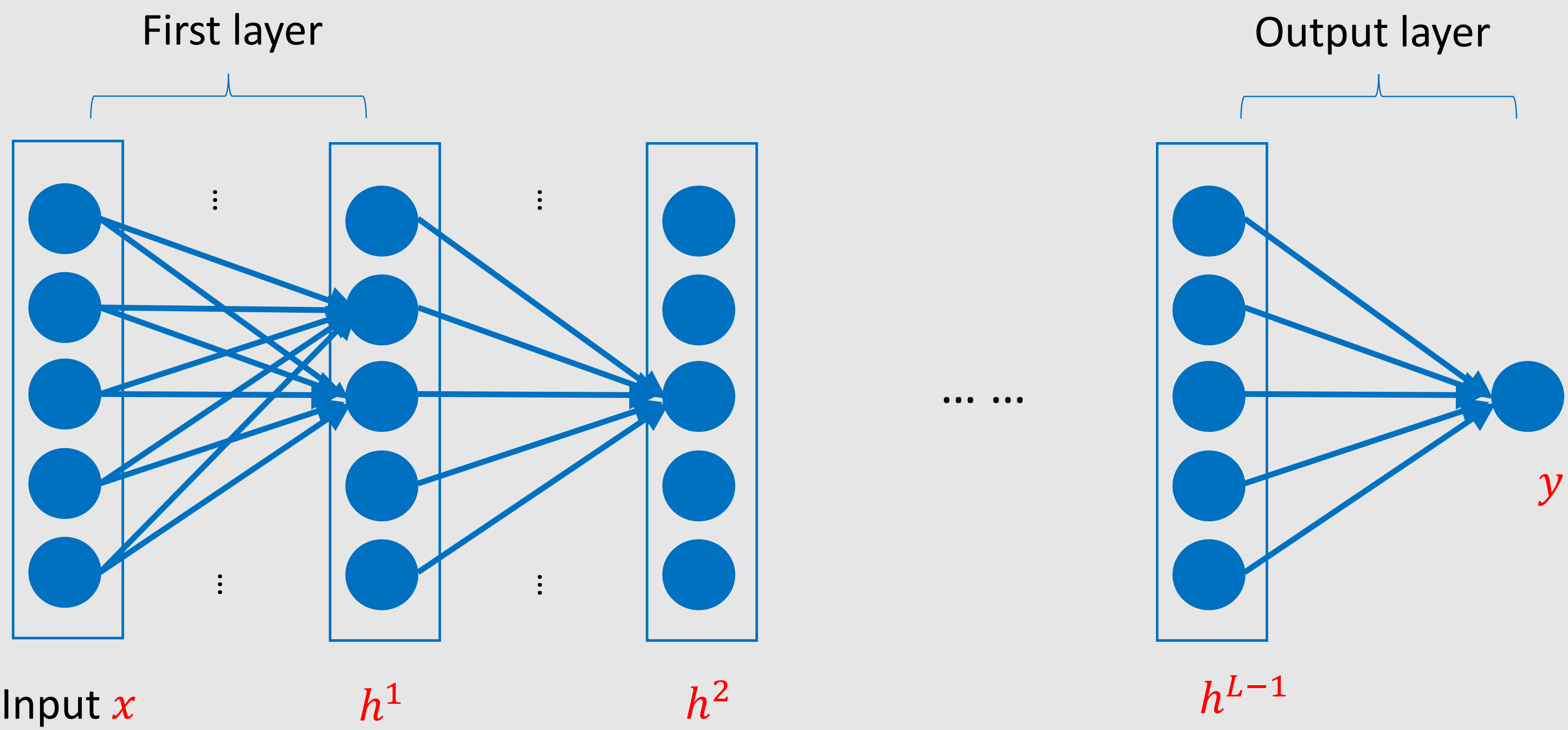
Theoretical Foundations of Deep Learning: Challenges

CS 839@UW-Madison

Yingyu Liang

Key Engine Behind Recent Success

- Deep Neural Networks: $y = f(x)$

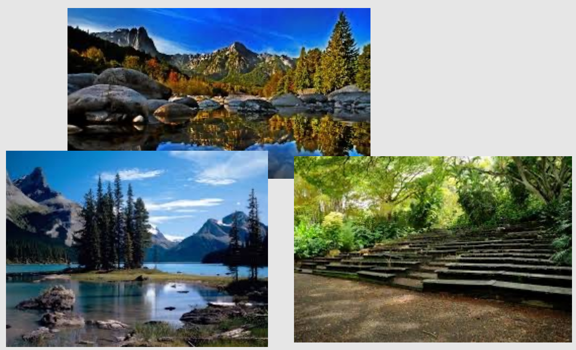


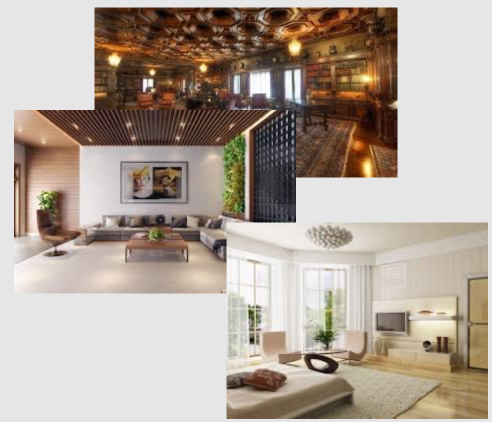
$h^i = \sigma\left(W_i h^{i-1}\right)$, with ReLU activation $\sigma(z) = \max(0, z)$

Key Engine Behind Recent Success

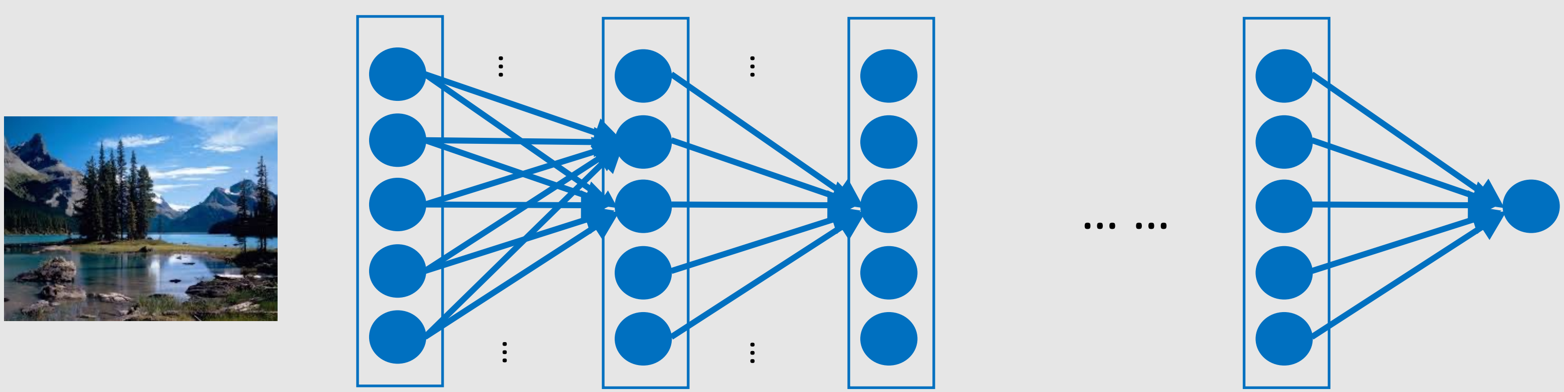
- Training Deep Neural Networks: $y = f(x; W)$
  - Given training data $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$
  - Try to find W such that the network fits the data

Outdoor





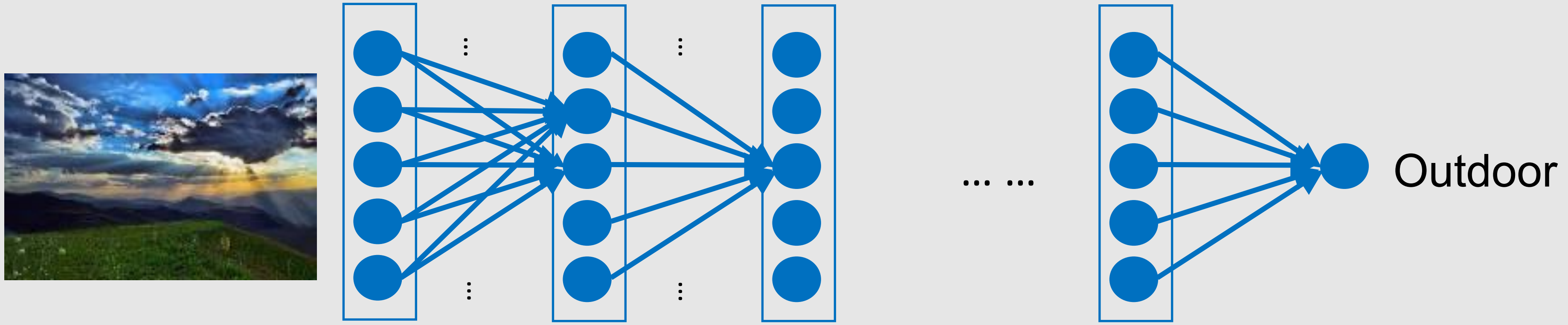
Indoor



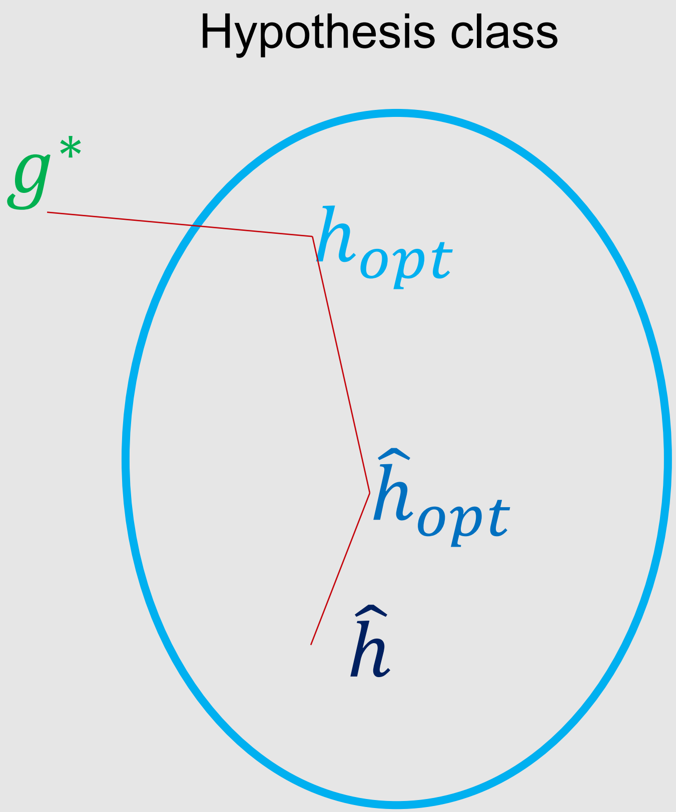
Outdoor

Key Engine Behind Recent Success

- Using Deep Neural Networks: $y = f(x; W)$
  - Given a new test point x
  - Predict $y = f(x; W)$



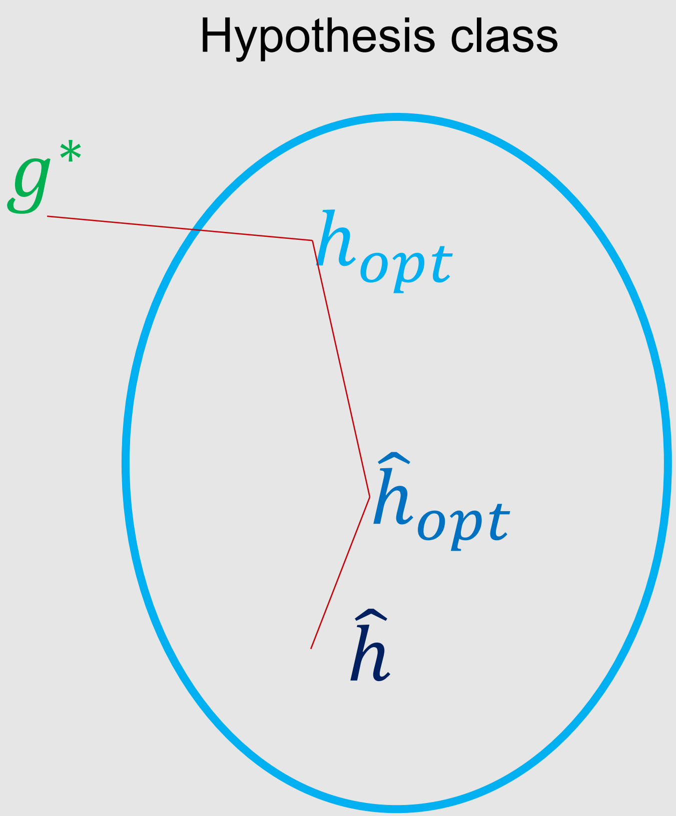
Risk Decomposition



- g^*: the ground-truth
- h_{opt}: the optimal hypothesis on the data distribution
- $\hat{h}_{opt}$: the optimal hypothesis on the training data
- $\hat{h}$: the trained hypothesis

The tradeoffs of large scale learning. Léon Bottou and Olivier Bousquet. Proceedings of the 20th International Conference on Neural Information Processing Systems, 2007.

Risk Decomposition



$$
\begin{aligned}
& R(\hat{h}) - R(g^*) \\
&= R(h_{opt}) - R(g^*) \\
&+ R(\hat{h}_{opt}) - R(h_{opt}) \\
&+ R(\hat{h}) - R(\hat{h}_{opt})
\end{aligned}
$$

Risk Decomposition

$$R(\hat{h}) - R(g^*)$$

Approximation error

$$= R(h_{opt}) - R(g^*)$$

Estimation error

$$+ R(\hat{h}_{opt}) - R(h_{opt})$$

Optimization error

$$+ R(\hat{h}) - R(\hat{h}_{opt})$$

Risk Decomposition

- Representation power (approximation error)
- Generalization (estimation error)
- Optimization (optimization error)

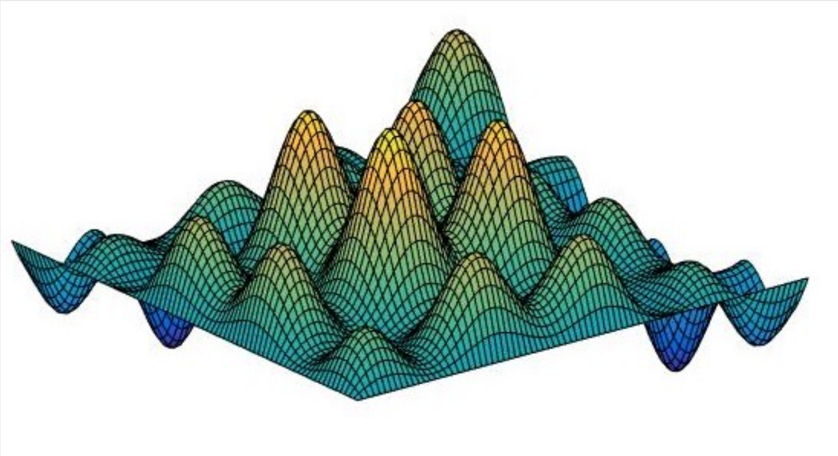
Fundamental Questions

- Optimization:

  Why can find W with good accuracy on training data?
- Generalization:

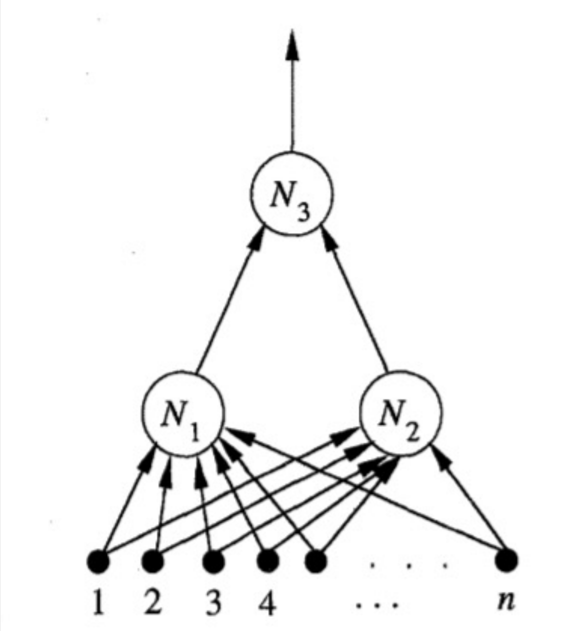
  Why the network also accurate on new test instances?

- First key challenge: the optimization is non-convex



Empirical Success v.s. Theoretical Hardness

- Theoretically hard
  - Training a 3-Node Neural Network is NP-Complete [Blum & Rivest, 93]



Training a 3-node neural network is NP-complete. Avrim Blum, and Ronald Rivest. Neural Networks 1992.

Empirical Success v.s. Theoretical Hardness

- Practically quite feasible
  - Simple algorithms like SGD often find good solutions
  - Practical networks are often very large and deep: hundreds of layers, thousands of nodes per layer

[1]Inception 5 (GoogLeNet)

Inception 7a

[1]Going Deeper with Convolutions, [C. Szegedy et al, CVPR 2015]

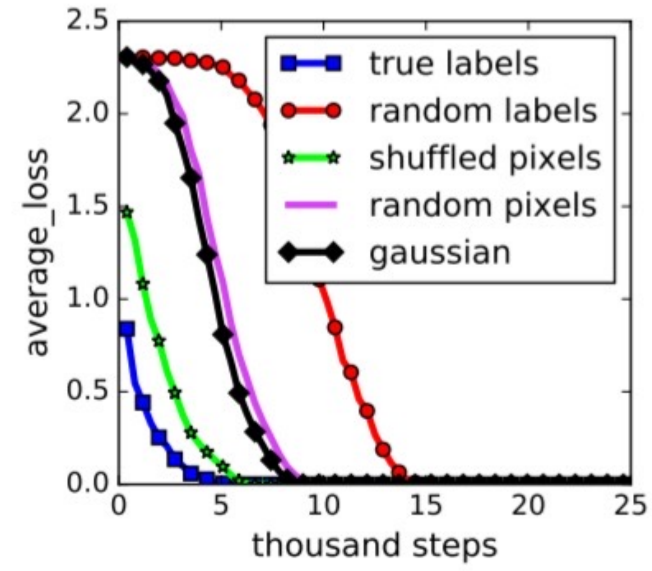
Key Challenge: Optimization

- Optimization lies in the center of many mysteries
- Empirical success v.s. theoretical hardness
- Overparameterized networks still good, contrast to traditional theory
  - So even if we assume optimization can be done, still cannot explain the good generalization performance
  - Optimization & generalization interweave with each other for NN learning

DNNs Easily Fit Random Labels

- Empirical observation: practical DNNs easily fit random labels

- First replace the training labels with random labels
- Then train with net architectures and methods used in practice



(a) learning curves

Understanding deep learning requires rethinking generalization. Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, Oriol Vinyals. ICLR 2017.

DNNs Easily Fit Random Labels

- Empirical observation: practical DNNs easily fit random labels

Surprising implications:

1. Practical DNNs are overparameterized
   - Sufficient to fit random labels → sufficient to fit labels with structure

DNNs Easily Fit Random Labels

- Empirical observation: practical DNNs easily fit random labels

Surprising implications:

1. Practical DNNs are overparameterized
2. Even optimization on random labels remains easy
   - Simple methods (variants of SGD) can converge to 0 (global optima)

DNNs Easily Fit Random Labels

- Empirical observation: practical DNNs easily fit random labels

Surprising implications:

1. Practical DNNs are overparameterized
2. Even optimization on random labels remains easy
3. Optimization automatically adapts to the structure of the data
    - With random labels, it fits the training labels by memorization (no generalization)
    - With practical labels with structure, it learns the underlying structure without memorization (good generalization)

DNNs Easily Fit Random Labels

- Empirical observation: practical DNNs easily fit random labels

Surprising implications:

1. Practical DNNs are overparameterized
2. Even optimization on random labels remains easy
3. Optimization automatically adapts to the structure of the data

- Appear to contradict traditional theory!