CPSC 440: Advanced Machine Learning

Neural Networks

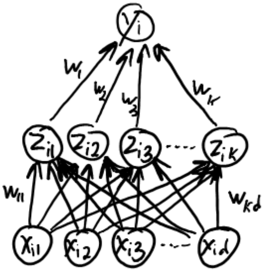
Mark Schmidt

University of British Columbia

Winter 2021

Last Time: Neural Networks

- In 340 we discussed feedforward neural networks for supervised learning.
- With 1 hidden layer the classic model has this structure:



- Motivation:
  - For some problems it's hard to find good features.
  - This learns features z that are good for particular supervised learning problem.
- Can be view as a DAG where latent variables z_c are deterministic.
  - Makes inference easy.

Neural Network Notation

- We'll continue using our supervised learning notation:

$$X = \begin{bmatrix} \text{——} & (x^1)^T & \text{——} \\ \text{——} & (x^2)^T & \text{——} \\ & \vdots & \\ \text{——} & (x^n)^T & \text{——} \end{bmatrix}, \quad y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^n \end{bmatrix},$$

- For the latent features and one hidden layer we'll use

$$Z = \begin{bmatrix} \text{——} & (z^1)^T & \text{——} \\ \text{——} & (z^2)^T & \text{——} \\ & \vdots & \\ \text{——} & (z^n)^T & \text{——} \end{bmatrix}, \quad v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{bmatrix}, \quad W = \begin{bmatrix} \text{——} & w_1 & \text{——} \\ \text{——} & w_2 & \text{——} \\ & \vdots & \\ \text{——} & w_k & \text{——} \end{bmatrix},$$

where Z is n by k and W is k by d.

Introducing Non-Linearity

- The obvious "linear-linear" model,

$$z^i = Wx^i, \quad \hat{y}^i = v^T z^i,$$

is degenerate since it's still a linear model.

- The classic solution is to introduce a non-linearity,

$$z^i = h(Wx^i), \quad \hat{y}^i = v^T z^i,$$

where a common-choice is applying sigmoid element-wise,

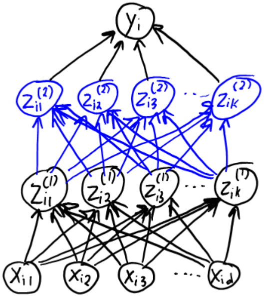
$$z_c^i = \frac{1}{1 + \exp(-w_c^T x^i)},$$

which is said to be the "activation" of neuron c on example i.

  - A universal approximator with k a function of n (also true for $\tanh$, ReLU, etc.)

Deep Neural Networks

- In deep neural networks we add multiple hidden layers,

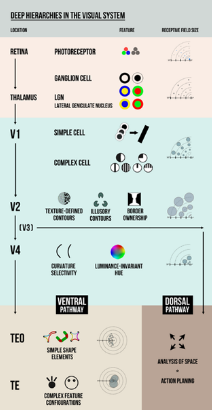


- Mathematically, with 3 hidden layers the classic model uses

$$\hat{y}^i = v^T \underbrace{h(W^3 \underbrace{h(W^2 \underbrace{h(W^1 x^i)}_{z^{i1}})}_{z^{i2}})}_{z^{i3}} \, .$$

Biological Motivation

- Deep learning is motivated by theories of deep hierarchies in the brain.



https://en.wikibooks.org/wiki/Sensory_Systems/Visual_Signal_Processing

- But most research is about making models work better, not be more brain-like.

Deep Neural Network History

- Popularity of deep learning has come in waves over the years.
  - Currently, it is one of the hottest topics in science.
- Recent popularity is due to unprecedented performance on some difficult tasks:
  - Speech recognition.
  - Computer vision.
  - Machine translation.
- These are mainly due to big datasets, deep models, and tons of computation.
  - Plus tweaks to classic models and focus on structured networks (CNNs, LSTMs).
- For a NY Times article discussing some of the history/successes/issues, see:

https://mobile.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html

Training Deep Neural Networks

- If we're training a network with 3 hidden layers and squared error, our objective is

$$f(v, W^1, W^2, W^3) = \frac{1}{2}\sum_{i=1}^{n}(\underbrace{v^T h(W^3 h(W^2 h(W^1 x^i)))}_{\hat{y}^i} - y^i)^2.$$

- Usual training procedure is stochastic gradient.
  - Highly non-convex and notoriously difficult to tune.
  - But we're discovering sets of tricks to make things easier to tune.
- Recent empirical/theoretical work indicates non-convexity may not be an issue:
  - Local minima found by SGD may be good for "large enough" networks.

Training Deep Neural Networks

- Some common data/optimization tricks we discussed in 340:
  - Data transformations.
    - For images, translate/rotate/scale/crop each x^i to make more data.
  - Data standardization: centering and whitening.
  - Adding bias variables.
  - Parameter initialization: "small but different", standardizing within layers.
  - Step-size selection: "babysitting", Bottou trick, Adam.
  - Momentum: heavy-ball and Nesterov-style modifications.
  - Batch normalization: adaptive standardizing within layers.
  - ReLU: replacing sigmoid with $\max\{0, w_c^T x^i\}$.
    - Avoids gradients extremely-close to zero.

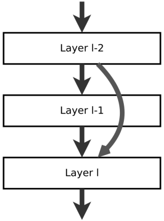
Training Deep Neural Networks

- Common forms tricks to fight overfitting:
  - Standard L2-regularization or L1-regularization "weight decay".
    - Sometimes with different λ for each layer.
    - Recent work shows this introduces bad local optima.
  - Early stopping of the optimization based on validation accuracy.
  - Dropout randomly zeroes z values to discourage dependence.
  - Implicit regularization from using SGD.
  - Hyper-parameter optimization to choose various tuning parameters.
    - "Neural architecture search": recent methods include search over graph structures.
  - Special architectures like convolutional neural networks:
    - Yields W^m that are very sparse and have many tied parameters.

"Residual" Networks (ResNets)

- Suppose we fit a deep neural network to a linearly-separable dataset.
  - Original features x are sufficient to perfectly classify training data.
  - For a deep neural network to work, each layer needs to preserve information in x.
    - You might be "wasting" parameters just re-representing data from previous layers.

- Consider residual networks:



https://en.wikipedia.org/wiki/Residual_neural_network

- Take a previous (non-transformed) layer as input to current layer.
  - Also called "skip connections" or "highway networks".

"Residual" Networks (ResNets)

- ResNets seemingly make learning easier:
  - You can "default" to just copying the previous layer.
  - The non-linear transform is only learning how to modify the input.
    - "Fitting the residual".
  - With ResNets, "you are done if problem is solved in any layer".
    - Because you can "skip" the effects of the remaining layers.

- This was a key idea behind first methods that used 100+ layers.
  - Easy for information about x to reach y through huge number of layers.
  - Won all tasks in ImageNet 2015 competition.
  - Evidence that biological networks have skip connections like this.

- Dense networks (DenseNets): connect to many previous layers.
  - Basically gets rid of vanishing gradient issue.

DenseNets

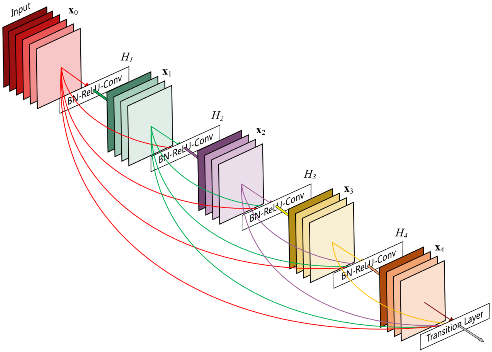


Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

https://arxiv.org/pdf/1608.06993.pdf

Pre-Training

- Suppose you want to solve a new object detection task.
  - Recognize a particular abnormality in radiology images.
- You only have a few labeled images, so is deep learning useless?

- An important concept in many computer vision applications is pre-training.
  - Learn new concepts faster by modifying networks trained on millions of images.
  - Uses that many "features" are common between tasks (edges, corners, shapes,...).

- Typical setup:
  - Take a network trained on ImageNet (typically VGG or ResNet).
  - Re-train the last layer to solve your problem (convex with usual losses).
- A form of transfer learning.
  - When you try to "transfer" information between learning problems.

Summary

- We overview many of the standard neural network tricks.
    - Multiple "layers" of hidden features.
    - Sigmoid or ReLU non-linear transformations.
    - SGD training, with lots of tricks/tuning.
    - Residual/skip connections.
    - Pre-training on related tasks with lots of data.
- Implicit regularization:
    - Some optimization methods may converge to regularized solutions.
- Next time: combining neural networks with the rest of the course.