Neural Networks: Practical Concerns

Machine Learning

Based on slides and material from Geoffrey Hinton, Richard Socher, Dan Roth, Yoav Goldberg, Shai Shalev-Shwartz and Shai Ben-David, and others

Neural Networks

- What is a neural network?
- Predicting with a neural network
- Training neural networks
- Practical concerns

This lecture

- What is a neural network?
- Predicting with a neural network
- Training neural networks
- Practical concerns

Practical concerns

1. Addressing problems with SGD
2. Preventing overfitting
3. Number of hidden layers

Training neural networks with SGD

- **No guarantee** of convergence**,** may oscillate or reach a local minima
- In practice, many large networks are trained on **large amounts of data** for realistic problems

Training neural networks with SGD

- **No guarantee** of convergence**,** may oscillate or reach a local minima
- In practice, many large networks are trained on **large amounts of data** for realistic problems
- Many epochs (tens of thousands) may be needed for adequate training
  - Large data sets may require many hours of CPU or GPU time
  - Sometimes specialized hardware even

Training neural networks with SGD

- **No guarantee** of convergence**,** may oscillate or reach a local minima
- In practice, many large networks are trained on **large amounts of data** for realistic problems
- Many epochs (tens of thousands) may be needed for adequate training
  - Large data sets may require many hours of CPU or GPU time
  - Sometimes specialized hardware even
- **Termination criteria**: Number of epochs, Threshold on training set error, No decrease in error, Increased error on a validation set

Training neural networks with SGD

- **No guarantee** of convergence**,** may oscillate or reach a local minima
- In practice, many large networks are trained on **large amounts of data** for realistic problems
- Many epochs (tens of thousands) may be needed for adequate training
  - Large data sets may require many hours of CPU or GPU time
  - Sometimes specialized hardware even
- **Termination criteria**: Number of epochs, Threshold on training set error, No decrease in error, Increased error on a validation set
- To **avoid local minima**: several trials with different random initial weights with majority or voting techniques

Preventing overfitting

- Running too many epochs may *over-train* the network and result in over-fitting

Preventing overfitting

- Running too many epochs may *over-train* the network and result in over-fitting

- Keep a **hold-out validation set** and test accuracy after every epoch

Preventing overfitting

- Running too many epochs may *over-train* the network and result in over-fitting
- Keep a **hold-out validation set** and test accuracy after every epoch
- Maintain weights for best performing network on the validation set and return it when performance decreases significantly beyond that

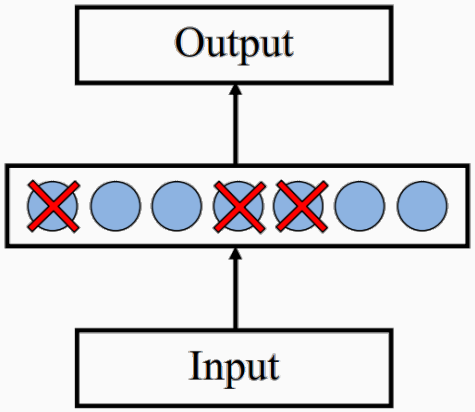
Preventing overfitting

- Running too many epochs may *over-train* the network and result in over-fitting
- Keep a **hold-out validation set** and test accuracy after every epoch
- Maintain weights for best performing network on the validation set and return it when performance decreases significantly beyond that
- To avoid losing training data to validation:
  - Use k-fold cross-validation to determine the average number of epochs that optimizes validation performance
  - Train on the full data set using this many epochs to produce the final results

Avoiding overfitting with Dropout training

Hinton et al, 2012

- During training, for each step, decide whether to delete a hidden unit with some probability *p*
  - That is, make predictions using only a randomly chosen set of neurons
  - Update only these neurons



- Tends to avoid overfitting
- Has a model averaging effect
  - Only some parameters get trained at any step

Number of hidden units

- **Too few hidden units** prevent the system from adequately fitting the data and learning the concept.
- **Using too many hidden units** leads to over-fitting.
- Similar cross-validation method can be used to determine an appropriate number of hidden units.

Neural networks: What we saw

- What is a neural network?
  - Multiple layers
    - Inner layers learn a **representation** of the data
  - Highly expressive
    - Is this always a good thing? What about the VC dimension? Overfitting?

- Training neural networks
  - Backpropagation

What we did not see

- Vast area, fast moving
  - Many new algorithms and tricks for learning that tweak on the basic gradient method
  - Eg: momentum, AdaDelta, Adam
- Some named neural networks
  - Restricted Boltzmann machines and autoencoders: Learn a latent representation of the data
  - Convolutional neural network: Modeled after the mammalian visual cortex, currently the state of the art for object recognition tasks
  - Recurrent neural networks: predict sequences
  - ...And many many more