Today: Outline

- **Revision**

- Reminders:
  - Problem Set 1, late submissions due: Oct 15 by midnight
  - Problem Set 1 model solution will be posted tomorrow
  - Midterm Exam, in class, Oct 20
  - Practice problems are posted

Q1a. Linear Regression: Used Car Dealership

- Imagine you work for a large online used car dealership and your boss would like you to estimate the price y (in dollars) the dealer should charge for a car based on the following features: x_1= car manufacturer, x_2= model, x_3= distance driven in miles, x_4= age in years, and binary features x_5= has side airbags, x_6= has leather seats, etc. For example, a feature vector for the i^{th} car could be $x^{(i)} = [4, 8, 17000, 5, 1, 0, \dots]$ where manufacturer and model are represented as integers. You have collected data points from previous car sales, $(x^{(i)}, y^{(i)}),\ i = 1, \dots, m$.
- Suppose you use gradient descent. How can you tell if it is converging?
- By plotting the cost as a function of the number of iterations: convergence is likely when the decrease in cost diminishes.

Q1b. Linear Regression: Used Car Dealership

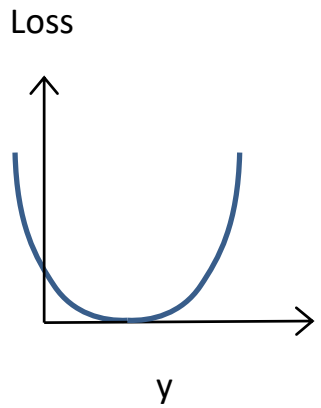
- Imagine you work for a large online used car dealership and your boss would like you to estimate the price y (in dollars) the dealer should charge for a car based on the following features: x_1= car manufacturer, x_2= model, x_3= distance driven in miles, x_4= age in years, and binary features x_5= has side airbags, x_6= has leather seats, etc. For example, a feature vector for the i^{th} car could be $x^{(i)} = [4, 8, 17000, 5, 1, 0, \dots]$ where manufacturer and model are represented as integers. You have collected data points from previous car sales, $(x^{(i)}, y^{(i)})$, $i = 1, \dots, m$.
- You find that your test accuracy is low. Name two things you can try to improve the result of linear regression without collecting any additional features.
- 1) Try a larger value for the step size;
- 2) In this case, some of the features have larger scale which could cause slow convergence, so we could use feature re-scaling to normalize all features, e.g. to [-1 1]

Q1c. Linear Regression: Used Car Dealership

- Imagine you work for a large online used car dealership and your boss would like you to estimate the price y (in dollars) the dealer should charge for a car based on the following features: x_1= car manufacturer, x_2= model, x_3= distance driven in miles, x_4= age in years, and binary features x_5= has side airbags, x_6= has leather seats, etc. For example, a feature vector for the i^{th} car could be $x^{(i)} = [4, 8, 17000, 5, 1, 0, \dots]$ where manufacturer and model are represented as integers. You have collected data points from previous car sales, $(x^{(i)}, y^{(i)}),\ i = 1, \dots, m$.
- You decide to add new features to improve your predictor. Is it a good idea to add distance driven in kilometers? Why or why not?
- No, because it is redundant with (a constant multiple of) the distance in miles and would not add new information.

Q1d. Linear Regression: Used Car Dealership

- Imagine you work for a large online used car dealership and your boss would like you to estimate the price y (in dollars) the dealer should charge for a car based on the following features: x_1= car manufacturer, x_2= model, x_3= distance driven in miles, x_4= age in years, and binary features x_5= has side airbags, x_6= has leather seats, etc. For example, a feature vector for the i^{th} car could be $x^{(i)} = [4, 8, 17000, 5, 1, 0, \dots]$ where manufacturer and model are represented as integers. You have collected data points from previous car sales, $(x^{(i)}, y^{(i)}),\ i = 1, \dots, m$.
- Typically in regression we minimize a square loss function, shown below. Does it make sense in this case? Why or why not?
- Yes, it makes sense, because it penalizes predicted values that are either below or above the actual sale price of the training example. However, as the next question suggests, there may be a better loss function we could use.

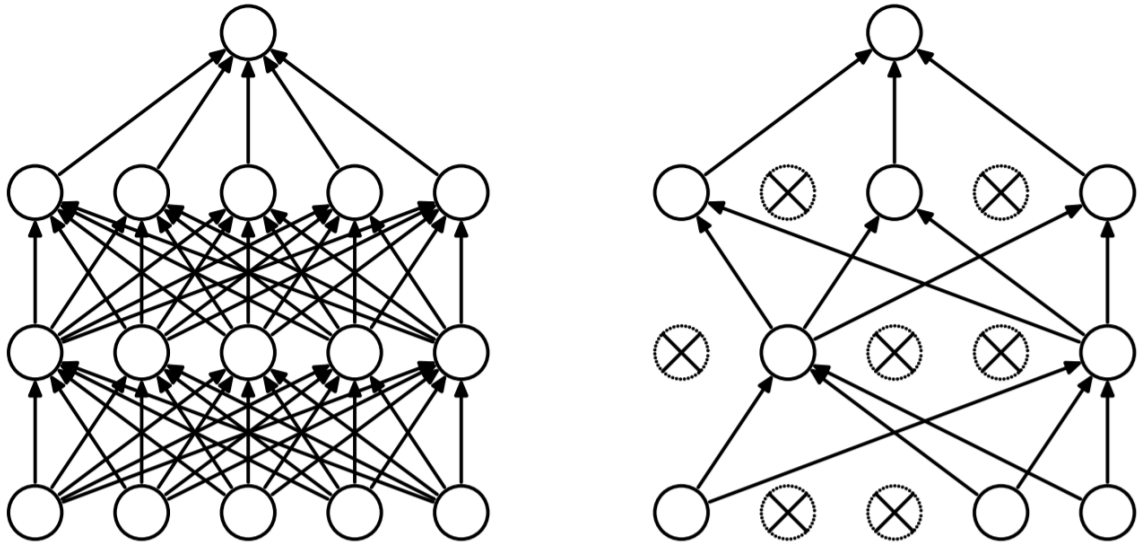


Q1e. Linear Regression: Used Car Dealership

- Imagine you work for a large online used car dealership and your boss would like you to estimate the price y (in dollars) the dealer should charge for a car based on the following features: x_1= car manufacturer, x_2= model, x_3= distance driven in miles, x_4= age in years, and binary features x_5= has side airbags, x_6= has leather seats, etc. For example, a feature vector for the i^{th} car could be $x^{(i)} = [4, 8, 17000, 5, 1, 0, \dots]$ where manufacturer and model are represented as integers. You have collected data points from previous car sales, $(x^{(i)}, y^{(i)}),\ i = 1, \dots, m$.
- Suppose you trained your model and it predicted a very low price for a particular Honda, $139. You check your training data, and find that all prices in the training examples are reasonable. The input features also look reasonable. What could be the reason for such a low prediction? How could you address it?
- Overfitting, which could be addressed by adding regularization to the model.

Q2. When is dropout applied? How does it help avoid overfitting?

- Many Deep Models employ dropout ***at training time*** to avoid overfitting, allowing for better generalization.



(a) Standard Neural Net

(b) After applying dropout.

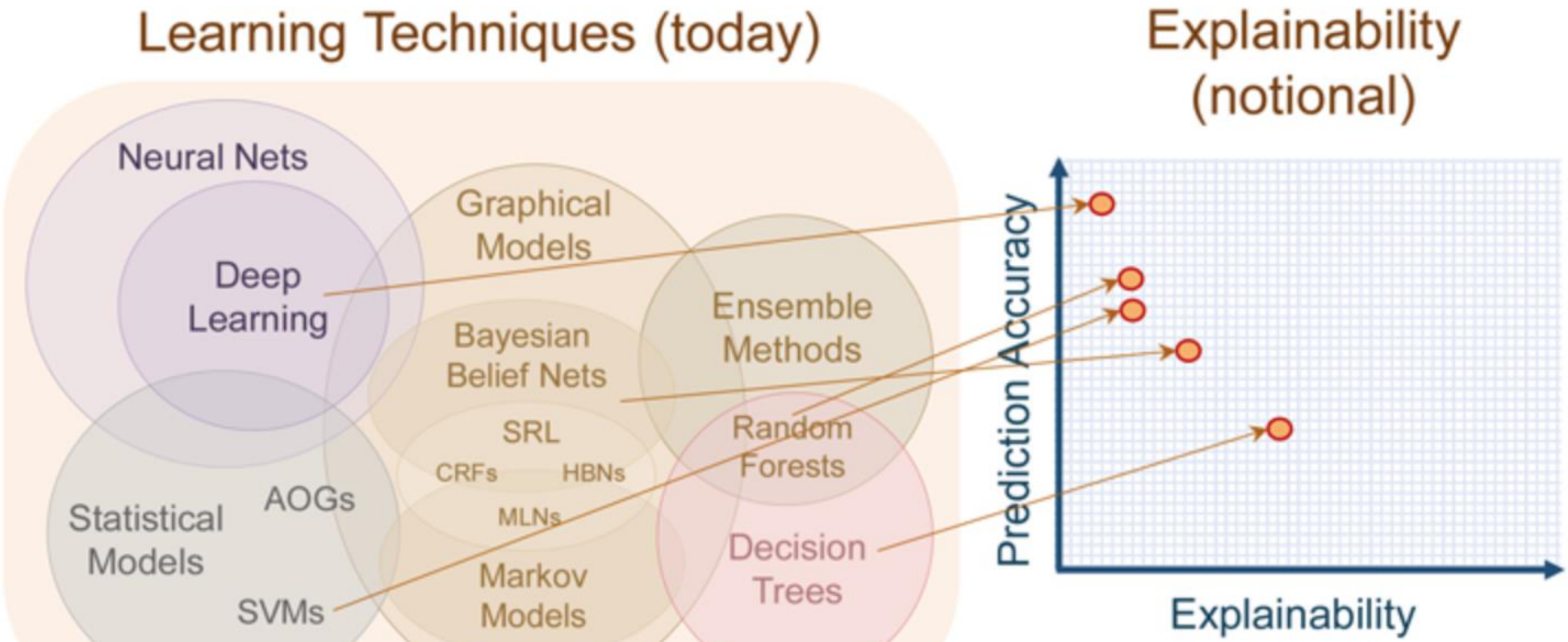
- By sampling the architecture several times, model averaging comes into effect that helps avoid overfitting.

[Srivastava *et al.*]

Q3. List an example of unfairness in AI and a proposed solution to make it more fair

- Data (input)
  - e.g. more arrests where there are more police
  - ***try to "correct" such data bias***
- Models (output)
  - e.g. discriminatory treatment of subpopulations
  - ***equality of false positive/negative rates; calibration***
- Algorithms (process)
  - e.g. don't learn outcomes of denied mortgages
  - ***design algorithms that are fair***

Q4. Describe the *Accuracy vs. Explainability* tradeoff in Machine Learning



Q5. How do we find the model parameters that result in a “best-fit” line for linear regression?

- These should be the parameter values that minimize the linear regression cost function:

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

- i.e. the $\theta's\ that\ minimize\ \ J(\theta_0, \theta_1, \ldots, \theta_n)$, written formally as:

$$\underset{\theta_0, \theta_1, \ldots, \theta_n}{\text{argmin}} \ \ J(\theta_0, \theta_1, \ldots, \theta_n)$$

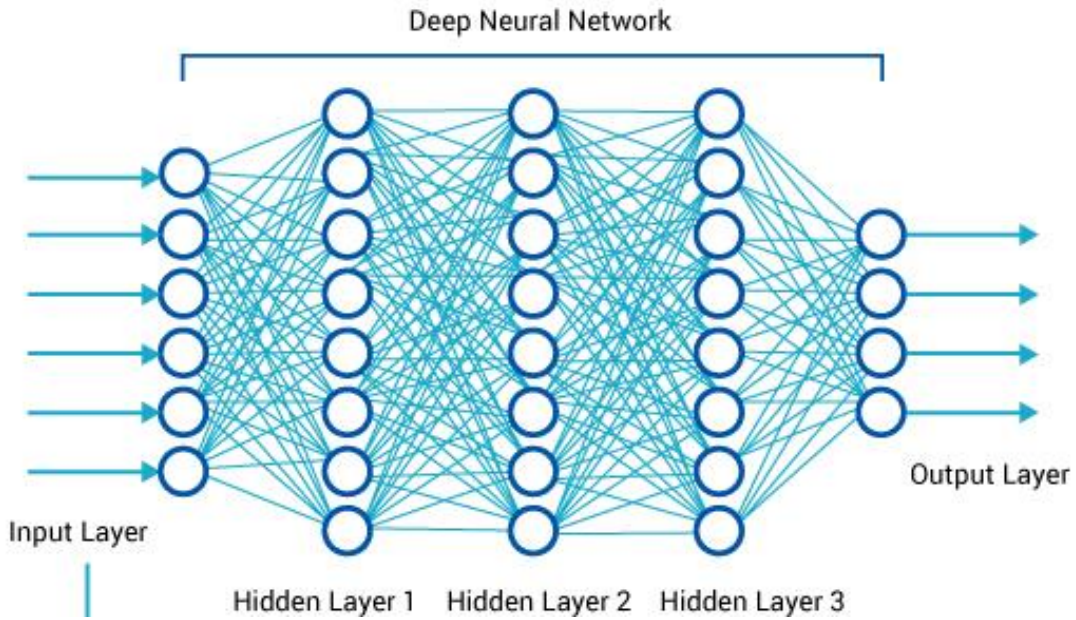
Q6. Write the formulation of a widely used cost function for classification, defining all its terms.

- Cross-entropy

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

Q7. What defines a *deep* neural network?

- A deep neural network is one that has more than one hidden layer.



Q8. Which data items are typically used to compute $\frac{\partial J(W)}{\partial W}$ in Gradient Descent?

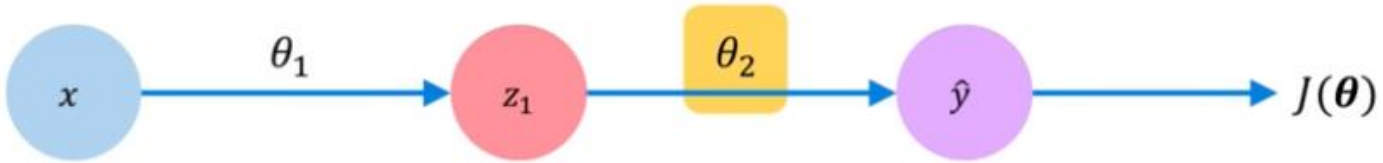
Gradient Descent

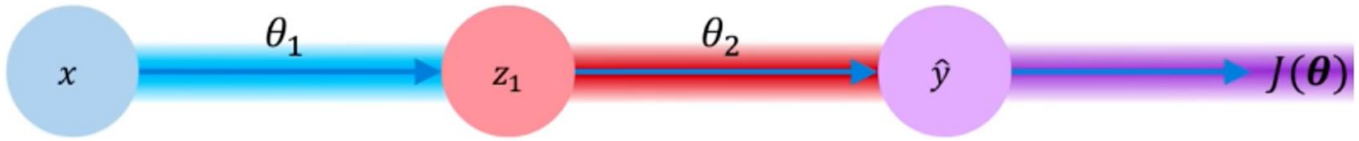
Algorithm

1. Initialize weights randomly $\sim \mathcal{N}(0, \sigma^2)$
2. Loop until convergence:
3. Compute gradient $\frac{\partial J(W)}{\partial W}$
4. Update weights, $W \leftarrow W - \eta \frac{\partial J(W)}{\partial W}$
5. Return weights

Compute over a mini-batch

Q9. Write mathematical formulation for "How does a small change in $\boldsymbol{\theta_1}$ affect the loss $\boldsymbol{J(\theta)}$?"



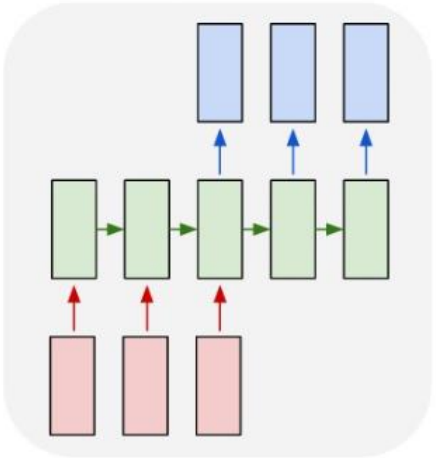


$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_1} = \frac{\partial J(\boldsymbol{\theta})}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial z_1} * \frac{\partial z_1}{\partial \theta_1}$$

Q10. Describe an application that would use the following many-to-many recurrent neural net

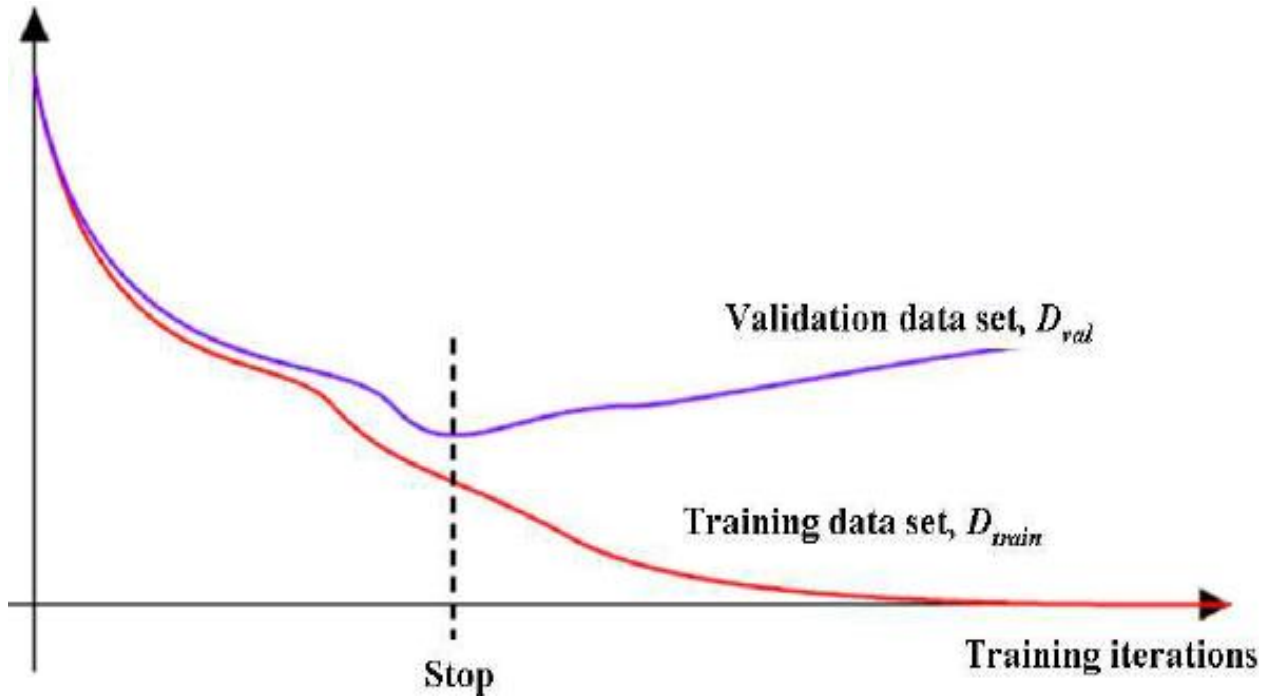
- Machine Translation

French Translation

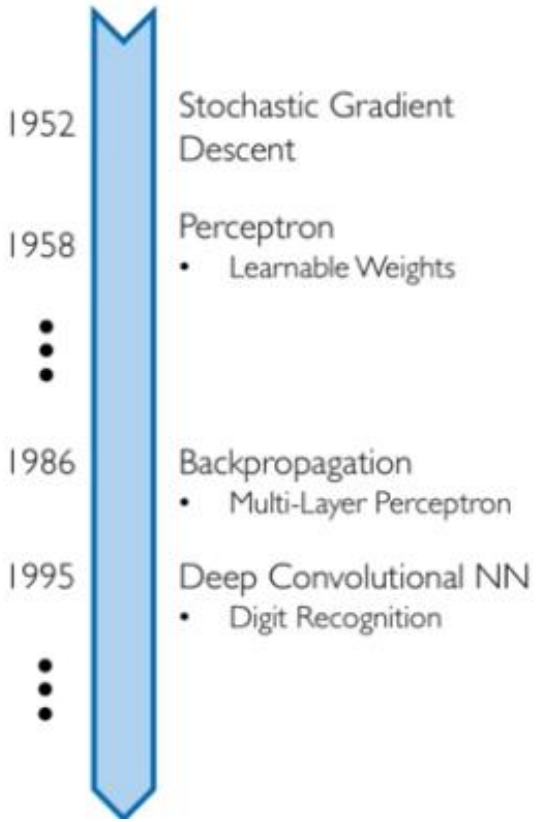


English Sentence

Q11. Draw a figure that would illustrate when you would perform early stopping to avoid overfitting



Q12. Why did deep learning only succeed in 2012?



Neural Networks date back decades, so why the resurgence?

1. Big Data

- Larger Datasets
- Easier Collection & Storage

IMAGENET

WIKIPEDIA

2. Hardware

- Graphics Processing Units (GPUs)
- Massively Parallelizable

3. Software

- Improved Techniques
- New Models
- Toolboxes

TensorFlow