EE 628
Deep Learning
Fall 2019

Lecture 2

09/05/2019

Assistant Prof. Sergul Aydore

Department of Electrical and Computer Engineering

Linear Neural Networks

- We need to cover the basics of neural network training

Linear Neural Networks

- We need to cover the basics of neural network training
- Entire training process includes

Linear Neural Networks

- We need to cover the basics of neural network training
- Entire training process includes
  - Defining simple neural network architectures
  - Handling data
  - Specifying loss functions
  - Training the model

Linear Regression

- **Linear**: the output is a linear combination of the input features

Linear Regression

- **Linear**: the output is a linear combination of the input features
- Example: estimating a price of a house (in dollars) based on area (in square feet) and age (in years)

Linear Regression

- **Linear**: the output is a linear combination of the input features
- Example: estimating a price of a house (in dollars) based on area (in square feet) and age (in years)
    - Our model can be expressed as:

$$\mathrm{price} = w_{\mathrm{area}} \times \mathrm{area} + w_{\mathrm{age}} \times \mathrm{age} + b$$

Linear Regression

- **Linear**: the output is a linear combination of the input features
- Example: estimating a price of a house (in dollars) based on area (in square feet) and age (in years)
  - Our model can be expressed as:

$$\mathrm{price} = w_{\mathrm{area}} \times \mathrm{area} + w_{\mathrm{age}} \times \mathrm{age} + b$$

- In the case of d variables: $\hat{y} = w_1 \times x_1 + \cdots + w_d \times x_d + b$
  - In vector form:

$$\hat{y} = \mathbf{w}^\top \mathbf{x} + b$$

Linear Regression

- **Linear**: the output is a linear combination of the input features
- Example: estimating a price of a house (in dollars) based on area (in square feet) and age (in years)
  - Our model can be expressed as:

$$\mathrm{price} = w_{\mathrm{area}} \times \mathrm{area} + w_{\mathrm{age}} \times \mathrm{age} + b$$

- In the case of d variables: $\hat{y} = w_1 \times x_1 + \cdots + w_d \times x_d + b$
  - In vector form:

$$\hat{y} = \mathbf{w}^\top \mathbf{x} + b$$

- For a collection of data points $\mathbf{X}$, the prediction can be expressed as:

$$\hat{\mathbf{y}} = \underbrace{\mathbf{X}}_{\text{Design matrix}}\mathbf{w} + \mathbf{b}$$

Training Data

- We need a collection of examples for which we know both the actual selling price of each house as well as their corresponding area and age.

Training Data

- We need a collection of examples for which we know both the actual selling price of each house as well as their corresponding area and age.
- Goal: minimize the error between the predicted price and the real price

Training Data

- We need a collection of examples for which we know both the actual selling price of each house as well as their corresponding area and age.
- Goal: minimize the error between the predicted price and the real price
- A house comprises one *sample*:
  - Its actual selling price is called a *label*
  - The factors used to predict the price are called *features*

Training Data

- We need a collection of examples for which we know both the actual selling price of each house as well as their corresponding area and age.
- Goal: minimize the error between the predicted price and the real price
- A house comprises one *sample*:
  - Its actual selling price is called a *label*
  - The factors used to predict the price are called *features*
- n: number of samples in our dataset

Training Data

- We need a collection of examples for which we know both the actual selling price of each house as well as their corresponding area and age.
- Goal: minimize the error between the predicted price and the real price
- A house comprises one *sample*:
  - Its actual selling price is called a *label*
  - The factors used to predict the price are called *features*
- n: number of samples in our dataset
- We index the samples by i:
  - Each input data point: $x^{(i)} = [x_1^{(i)}, x_2^{(i)}]$
  - The corresponding label: $y^{(i)}$

Loss Function

- Need to measure error between the predicted value and the real value of the price: i.e. *loss function*

Loss Function

- Need to measure error between the predicted value and the real value of the price: i.e. *loss function*
- The smaller the value, the smaller the error

Loss Function

- Need to measure error between the predicted value and the real value of the price: i.e. *loss function*
- The smaller the value, the smaller the error
- A common choice: the *square loss*

Loss Function

- Need to measure error between the predicted value and the real value of the price: i.e. *loss function*
- The smaller the value, the smaller the error
- A common choice: the *square loss*
- For given parameters $\mathbf{w}$ and b, we can express the error:

$$l^{(i)}(\mathbf{w}, b) = \frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

Loss Function

- Need to measure error between the predicted value and the real value of the price: i.e. *loss function*
- The smaller the value, the smaller the error
- A common choice: the *square loss*
- For given parameters $\mathbf{w}$ and b, we can express the error:

$$l^{(i)}(\mathbf{w}, b) = \frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

- To measure the quality of a model on the entire dataset, we average the losses on the training set.

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right)^2$$

Loss Function

- Need to measure error between the predicted value and the real value of the price: i.e. *loss function*
- The smaller the value, the smaller the error
- A common choice: the *square loss*
- For given parameters $\mathbf{w}$ and b, we can express the error:

$$l^{(i)}(\mathbf{w}, b) = \frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

- To measure the quality of a model on the entire dataset, we average the losses on the training set.

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right)^2$$

- We want to find parameters that minimize the average loss across all training samples

$$\mathbf{w}^*, b^* = \arg\min_{\mathbf{w}, b} L(\mathbf{w}, b)$$

Analytic Solution

- Linear regression is unusually a simple optimization problem

Analytic Solution

- Linear regression is unusually a simple optimization problem
- There is just one critical point on the loss surface corresponding to the global minimum.

Analytic Solution

- Linear regression is unusually a simple optimization problem
- There is just one critical point on the loss surface corresponding to the global minimum.
- Taking the derivative of the loss with respect to $\mathbf{w}$, and setting it equal to 0 gives the analytic solution:

$$\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

Gradient Descent

- The key trick behind nearly all deep learning is gradient descent:
  - To reduce the error gradually by iteratively updating the parameters

Follow the slope!

Credit: http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture03.pdf

Gradient Descent

- The key trick behind nearly all deep learning is gradient descent:
  - To reduce the error gradually by iteratively updating the parameters
- On convex loss surfaces it will eventually converge to a global minimum

Follow the slope!

Credit: http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture03.pdf

Gradient Descent

- The key trick behind nearly all deep learning is gradient descent:
  - To reduce the error gradually by iteratively updating the parameters
- On convex loss surfaces it will eventually converge to a global minimum
- For nonconvex surfaces, it will at least lead towards a (hopefully good) local minimum

Follow the slope!

Credit: http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture03.pdf

The Normal Distribution and Squared Loss

- Recall the normal distribution with mean μ and variance σ^2: $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$

The Normal Distribution and Squared Loss

- Recall the normal distribution with mean μ and variance σ^2: $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$
- The key assumption in linear regression with least mean squares loss is that the observations actually arise from noisy observations: $y = \mathbf{w}^\top \mathbf{x} + b + \epsilon \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2)$

The Normal Distribution and Squared Loss

- Recall the normal distribution with mean μ and variance σ^2: $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$
- The key assumption in linear regression with least mean squares loss is that the observations actually arise from noisy observations: $y = \mathbf{w}^\top \mathbf{x} + b + \epsilon \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2)$
- This allows us to write out the likelihood of seeing a particular y for a given data $\mathbf{x}$:

$$p(y|\mathbf{x}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2}(y - \mathbf{w}^\top \mathbf{x} - b)^2\right)$$

The Normal Distribution and Squared Loss

- Recall the normal distribution with mean μ and variance σ^2: $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$
- The key assumption in linear regression with least mean squares loss is that the observations actually arise from noisy observations: $y = \mathbf{w}^\top \mathbf{x} + b + \epsilon \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2)$
- This allows us to write out the likelihood of seeing a particular y for a given data $\mathbf{x}$:

$$p(y|\mathbf{x}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2}(y - \mathbf{w}^\top \mathbf{x} - b)^2\right)$$

- A good way of finding the most likely values of b and w is to maximize the likelihood of the entire dataset $p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} p(y^{(i)}|\mathbf{x}^{(\mathbf{i})})$

The Normal Distribution and Squared Loss

- Recall the normal distribution with mean μ and variance σ^2: $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$
- The key assumption in linear regression with least mean squares loss is that the observations actually arise from noisy observations: $y = \mathbf{w}^\top \mathbf{x} + b + \epsilon \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2)$
- This allows us to write out the likelihood of seeing a particular y for a given data $\mathbf{x}$:

$$p(y|\mathbf{x}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2}(y - \mathbf{w}^\top \mathbf{x} - b)^2\right)$$

- A good way of finding the most likely values of b and w is to maximize the likelihood of the entire dataset $p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} p(y^{(i)}|\mathbf{x}^{(\mathbf{i})})$
- The estimators found by maximizing the likelihood of the data is known as *Maximum likelihood estimators.*

The Normal Distribution and Squared Loss

- Recall the normal distribution with mean μ and variance σ^2: $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$
- The key assumption in linear regression with least mean squares loss is that the observations actually arise from noisy observations: $y = \mathbf{w}^\top \mathbf{x} + b + \epsilon \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2)$
- This allows us to write out the likelihood of seeing a particular y for a given data $\mathbf{x}$:

$$p(y|\mathbf{x}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2}(y - \mathbf{w}^\top \mathbf{x} - b)^2\right)$$

- A good way of finding the most likely values of b and w is to maximize the likelihood of the entire dataset $p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} p(y^{(i)}|\mathbf{x}^{(\mathbf{i})})$
- The estimators found by maximizing the likelihood of the data is known as *Maximum likelihood estimators.*
- Maximizing the product of many exponential functions is awkward. A much better way is to minimize the Negative Log-likelihood- $\log p(\mathbf{y} \mid \mathbf{x})$:

$$-\log(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{n} \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^\top \mathbf{x}^{(i)} - b)^2$$

The Normal Distribution and Squared Loss

- Recall the normal distribution with mean μ and variance σ^2: $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$
- The key assumption in linear regression with least mean squares loss is that the observations actually arise from noisy observations: $y = \mathbf{w}^\top \mathbf{x} + b + \epsilon \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2)$
- This allows us to write out the likelihood of seeing a particular y for a given data $\mathbf{x}$:

$$p(y|\mathbf{x}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2}(y - \mathbf{w}^\top \mathbf{x} - b)^2\right)$$

- A good way of finding the most likely values of b and w is to maximize the likelihood of the entire dataset $p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} p(y^{(i)}|\mathbf{x}^{(\mathbf{i})})$
- The estimators found by maximizing the likelihood of the data is known as *Maximum likelihood estimators.*
- Maximizing the product of many exponential functions is awkward. A much better way is to minimize the Negative Log-likelihood- $\log p(\mathbf{y} \mid \mathbf{x})$:

$$-\log(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{n} \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(y^{(i)} - \mathbf{w}^\top \mathbf{x}^{(i)} - b)^2$$

- It follows that maximum likelihood in a linear model with additive Gaussian noise is equivalent to linear regression with squared loss

Implementation of linear regression from scratch

- Now, start your ipython notebook

Next

- Softmax regression
- Multilayer perceptron