Outline

• Review
  • Maximum Likelihood
  • Logistic Regression
  • Softmax Regression
Ingredients for a Neural Networks Algorithm

- Inputs
- Outputs
- Architecture
- **Objective Function**
- **Learning Rule**
Anatomy of Simplest Net

- Activation function
- Weighted sum

Inputs

Weights ($w$)

Output
Linear Regression

- Output: \( \hat{y} = Xw \in \mathbb{R}^n \)
- Weights: \( w \in \mathbb{R}^d \)
- Input: \( X \in \mathbb{R}^{n \times d} \)
- Minimize \( \mathcal{L}(y, \hat{y}) = ||y - \hat{y}||_2^2 \)
Gradient descent

• Start with initial (random) guess \( w_1 \)

• For \( T \) rounds, update

\[
\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} \ell (\mathbf{y}, \hat{\mathbf{y}})
\]

\[
\nabla_{\mathbf{w}} \ell (\mathbf{y}, \hat{\mathbf{y}}) = \frac{d \ell (\mathbf{y}, \hat{\mathbf{y}})}{d \mathbf{w}}
\]
Low-Quality Animation
Stochastic Gradient Descent

• For $t$ in $1 \ldots T$
  
  $t \in 1, \ldots T$

  Sample row $x_i \in x_1, \ldots x_n$

  Calculate prediction $\hat{y}_i = w^T x_i$

  Calculate loss $\ell(y_i, \hat{y}_i) = ||y_i - \hat{y}_i||_2^2$

  Apply update $w \leftarrow w - \eta \nabla_w \ell(y_i, \hat{y}_i)$
Gradient Calculation (Linear)

$$\nabla_w \ell(y_i, \hat{y}_i) = \frac{\partial}{\partial w} ||y_i - \hat{y}||^2_2$$

$$= \frac{\partial}{\partial w} (\hat{y} - y_i)^2$$

$$= 2(\hat{y} - y_i)x$$
Sanity Check

• Stochastic gradient is remarkably resilient
• Sometimes code will work even if there is a bug in gradient calculation
• But beware of sign errors
• How might you sanity check the sign?
Outline

- Review
- **Maximum Likelihood**
- Logistic Regression
- Softmax Regression
Maximum Liklihood Estimation

• So far, we haven’t discussed probability

• But it’s often useful to think of supervised learning as modeling a condition probability \( P(y|x) \)

• The task of learning is to learn a posterior probability \( P(w|D) \) for data \( D = (X, y) \)
Max Likelihood (cont’d)

• Learning a distribution over parameters is hard

• Maximum likelihood principle suggests we choose the parameters which maximize $P(D|\mathbf{w})$

$\mathbf{w}_{MLE} = \arg \max P(D|\mathbf{w})$

• But what does $P(D|\mathbf{w})$ really mean?
Gaussian Error Model

- We consider the prediction $\hat{y}$ to specify a Gaussian probability distribution centered at $\hat{y}$ and with variance $\sigma^2$

$$y = Xw + \epsilon$$
$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$
Maximum Likelihood Linear Regression

\[ P(y|\hat{y}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( \frac{-(y - \hat{y})^2}{2\sigma^2} \right) \]

\[ \mathbf{w}^{MLE} = \arg \max_{\mathbf{w}} P(y|\hat{y}) = \arg \max_{\mathbf{w}} \prod_{i=1}^{n} P(y_i|\hat{y}_i) \]

\[ = \arg \max_{\mathbf{w}} \log P(y|\hat{y}) \]

\[ = \arg \max_{\mathbf{w}} \sum_{i=1}^{n} \log P(y_i|\hat{y}_i) \]

\[ = \arg \min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \]
Outline

• Review

• Maximum Likelihood

• Logistic Regression

• Softmax Regression
Classification

- Sometimes we want to map data to categories (vs scalars)
- Example problems: spam detection, object recognition, medical diagnosis
Logistic Regression

For a Single Example:

- **Label:** \( y \in \{0, 1\} \)
- **Output:** \( \hat{y} = \sigma(w^T x) \)
- **Weights:** \( w \in \mathbb{R}^d \)
- **Input:** \( x \in \mathbb{R}^d \)
Sigmoid Activation

- Activation function is $\sigma(z) = \frac{1}{1 + e^{-z}}$
- Function *squashes* values to $[0, 1]$
- Can be interpreted as probabilities
MLE for Logistic Regression

• Again, we’re going to choose a maximum likelihood objective

• Interpret \( \hat{y} \) as \( P(y = 1) \) that and \( 1 - \hat{y} = P(y = 0) \)

• Choose \( w^{MLE} = \arg \max_w \log P(y|\hat{y}) \)

• Objective: Minimize negative log likelihood:

\[
\min - \sum_{i=1}^{n} y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)
\]
Train Logistic Regression

• Derive gradient  \( \nabla_w \ell(y, \hat{y}) = \frac{d\ell(y, \hat{y})}{dw} \)

• When  \( y_i = 1 \),  \( \ell = -\log \hat{y}_i = -\log \sigma(w^T x) \)

\[
\frac{\partial \ell}{w} = -\frac{1}{\sigma(w^T x)} \sigma(w^T x)(1 - \sigma(w^T x))x
\]

\[= (\hat{y} - y)x \]
Train Logistic Regression (2)

- Derive gradient $\nabla_w \ell(y, \hat{y}) = \frac{d\ell(y, \hat{y})}{dw}$

- When $y_i = 0$, $\ell = -\log 1 - \hat{y} = -\log 1 - \sigma(w^T x)$

$$\frac{\partial \ell}{\partial w} = -\frac{1}{1 - \sigma(w^T x)}(-1)\sigma(w^T x)(1 - \sigma(w^T x))x$$

$$= (\hat{y} - y)x$$
Apply SGD

- Same as for linear regression
Outline

• Review
• Maximum Likelihood
• Logistic Regression
• Softmax Regression
Multi-Class Classification

• What do we do when we have multiple classes?

• Might want to group email as \{inbox, social, updates, spam\}

• Object detection in images have thousands of categories

• This task is called multi-class classification
Softmax Regression

For a Single Example:

- Label: $y \in 1, ..., k$
- Output: $\hat{y} = \text{softmax}(x^T W)$
- Weights: $W \in \mathbb{R}^{d \times k}$
- Input: $x \in \mathbb{R}^d$
The Softmax Operation

• Now we have k outputs

• To use maximum likelihood, we would like an output that gives any label a probability

• The softmax operation enforces that our output is a valid multinomial distribution

\[
\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}
\]
Softmax Objective

- We represent our label as a one-hot over all the possible outputs

- A one hot has value one for true label, 0 elsewhere: 
  $[\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ ]$

- To perform maximum likelihood, we try to maximize the probability assigned to the correct class

- This loss function is called cross-entropy $-\sum_{k} y_k \log \hat{y}_{k}$