CSC 561: Neural Networks and Deep Learning

Gradient Descent

Marco Alvarez

Department of Computer Science and Statistics
University of Rhode Island

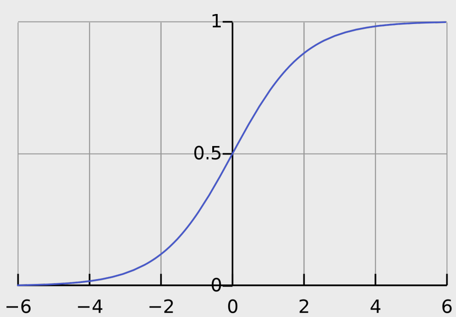
Spring 2024

THINK BIG WE DO℠

Logistic regression

Logistic function

$$\sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$$



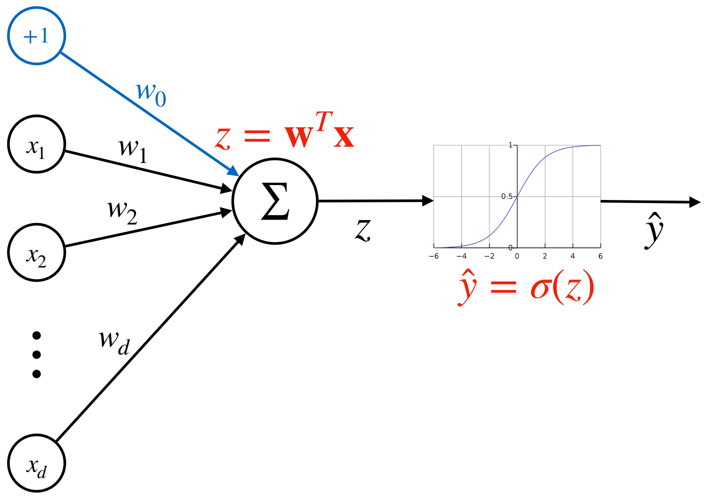
mapping ℝ to [0,1]

continuous and differentiable

Logistic regression

- Binary classifier
  - models $P(y \mid \mathbf{x}), \mathbf{x} \in \mathbb{R}^d, y \in \{+1, -1\}$
    - a threshold (e.g., $\theta = 0.5$) may be used for a final binary classification
  - uses the logistic function
- Considered a **linear classifier**
  - although the "activation function" is nonlinear

NN-like



Probabilistic interpretation

$P(y = -1 \mid \mathbf{x}; \mathbf{w}) = 1 - P(y = +1 \mid \mathbf{x}; \mathbf{w})$ $\qquad \sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$

(probability of class +1) $\quad P(y = +1 \mid \mathbf{x}; \mathbf{w}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}} = \sigma(\mathbf{w}^T\mathbf{x})$

(probability of class -1) $\quad P(y = -1 \mid \mathbf{x}; \mathbf{w}) = \frac{1}{1+e^{\mathbf{w}^T\mathbf{x}}} = \sigma(-\mathbf{w}^T\mathbf{x})$

$$P(y \mid \mathbf{x}; \mathbf{w}) = \frac{1}{1+e^{-y\mathbf{w}^T\mathbf{x}}} = \sigma(y\mathbf{w}^T\mathbf{x})$$

note that $P(y|\mathbf{x}) > 0.5$ when $y\mathbf{w}^T\mathbf{x} > 0$ (correct classifications)

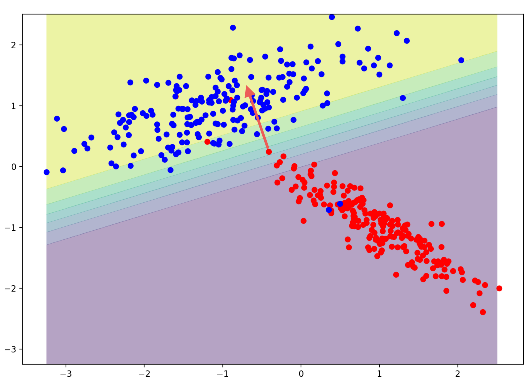
Linear decision boundary

$$P(y \mid \mathbf{x}; \mathbf{w}) = \frac{1}{1+e^{-y\mathbf{w}^T\mathbf{x}}} = \frac{1}{2}$$

$$1 + e^{-y\mathbf{w}^T\mathbf{x}} = 2$$

$$e^{-y\mathbf{w}^T\mathbf{x}} = 1$$

$$\mathbf{w}^T\mathbf{x} = 0$$



Learning the parameters

MLE

- Maximum likelihood estimation
  - choose parameters **w** that maximize the **conditional data likelihood** $P(\mathbf{y} \mid X; \mathbf{w})$, i.e., the probability of the observed values conditioned on the feature values

`i.i.d. assumption`

$$P(\mathbf{y} \mid X; \mathbf{w}) = \prod_{i=1}^{n} P(y^{(i)} \mid \mathbf{x}^{(i)}; \mathbf{w})$$

  - objective function:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \prod_{i=1}^{n} P(y^{(i)} \mid \mathbf{x}^{(i)}; \mathbf{w})$$

MLE

$$\begin{aligned}
\mathbf{w}^* &= \arg\max_{\mathbf{w}} \prod_{i=1}^{n} P(y^{(i)} \mid \mathbf{x}^{(i)}; \mathbf{w}) \\
&= \arg\max_{\mathbf{w}} \log\left(\prod_{i=1}^{n} P(y^{(i)} \mid \mathbf{x}^{(i)}; \mathbf{w})\right) \\
&= \arg\max_{\mathbf{w}} \frac{1}{n} \log\left(\prod_{i=1}^{n} P(y^{(i)} \mid \mathbf{x}^{(i)}; \mathbf{w})\right) \qquad \frac{1}{1 + e^{-y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)}}} \\
&= \arg\max_{\mathbf{w}} -\frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)}}\right)
\end{aligned}$$

negative log likelihood

$$= \arg\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \boxed{\log\left(1 + e^{-y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)}}\right)}$$

per instance loss

Applying gradient descent

`cross-entropy loss: no closed-form solution, but loss is convex`

$$J(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)}}\right)$$

gradient

$$\nabla_{\mathbf{w}} J(\mathbf{w}) = \left[\frac{\partial J(\mathbf{w})}{\partial w_0}, \ldots, \frac{\partial J(\mathbf{w})}{\partial w_d}\right]$$

$f(x) = \log(1 + e^z)$

$f'(x) = \frac{e^z}{1 + e^z}$

$= \sigma(z)$

$$\frac{\partial J(\mathbf{w})}{\partial w_j} = -\frac{1}{n} \sum_{i=1}^{n} \sigma\left(-y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)}\right) y^{(i)} x_j^{(i)}$$

Show me the code

```
# ...
```

Extension to multiple classes

From binary to C>2 classes

- Binary logistic regression

$$P(y=+1 \mid \mathbf{x}; \mathbf{w}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}} = \frac{e^{\mathbf{w}^T\mathbf{x}}}{e^{\mathbf{w}^T\mathbf{x}}+1}$$

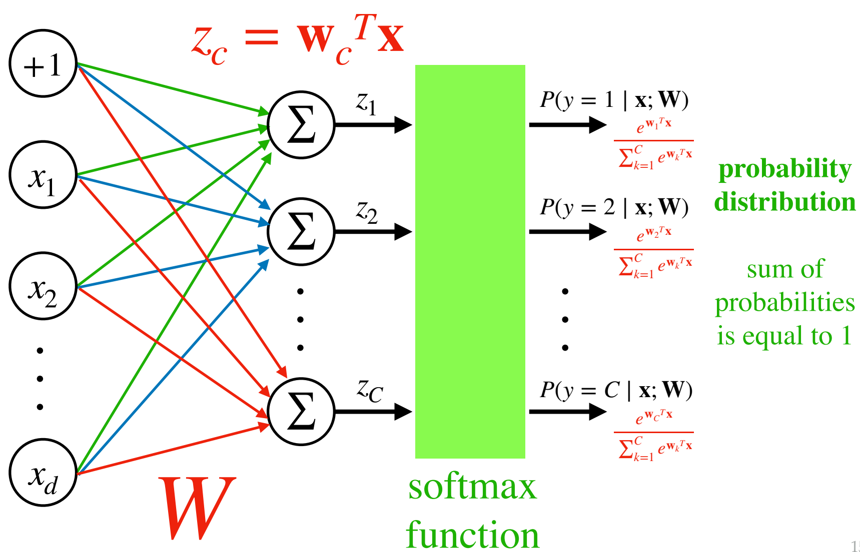
- C classes

softmax function

$$P(y=c \mid \mathbf{x}; W) = \boxed{\frac{e^{\mathbf{w_c}^T\mathbf{x}}}{\sum_{k=1}^{C} e^{\mathbf{w_k}^T\mathbf{x}}}}$$

$W_{C\times(d+1)}$ is a matrix where rows are "per-class" weight vectors

NN-like



Multinomial logistic regression

- Replace the **logistic** by the **softmax** function
  - predict the class with the highest probability score

$$\hat{y} = \arg\max_c P(y=c \mid \mathbf{x}; \mathbf{W})$$

- How to learn the parameters?
  - use MLE to derive a **loss function** … then apply **gradient descent**

MLE

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \frac{1}{n}\prod_{i=1}^{n} P(y^{(i)} \mid \mathbf{x}^{(i)}; \mathbf{W})$$

$$= \arg\max_{\mathbf{W}} \frac{1}{n}\prod_{i=1}^{n}\prod_{c=1}^{C} P\left(y^{(i)} = c \mid \mathbf{x}^{(i)}; \mathbf{W}\right)^{t_{i,c}}$$

Consider a matrix $T_{n\times C}$ where every row is a **one-hot encoding** of the target variable

$$= \arg\max_{\mathbf{W}} \frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C} t_{i,c}\log\left(P(y^{(i)} = c \mid \mathbf{x}^{(i)}; \mathbf{W})\right)$$

$$= \arg\min_{\mathbf{W}} -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C} t_{i,c}\log\left(P(y^{(i)} = c \mid \mathbf{x}^{(i)}; \mathbf{W})\right)$$

$$= \arg\min_{\mathbf{W}} \frac{1}{n}\sum_{i=1}^{n}\boxed{\sum_{c=1}^{C} - t_{i,c}\log\left(\frac{e^{\mathbf{w}_c\mathbf{x}^{(i)}}}{\sum_{k=1}^{C} e^{\mathbf{w}_k\mathbf{x}^{(i)}}}\right)}$$

per instance cross-entropy loss

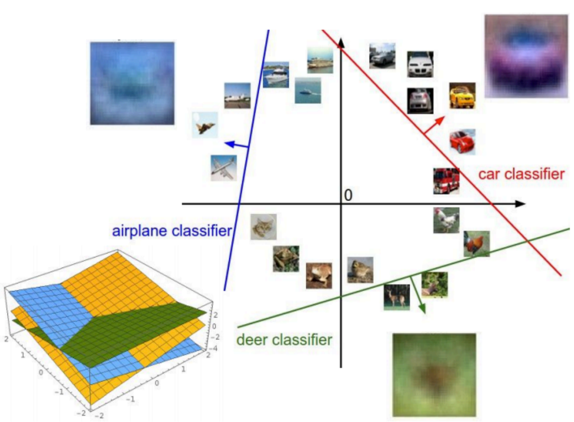
Softmax and the cross-entropy loss

	Unnormalized log-probabilities / logits		unnormalized probabilities (Probabilities must be >= 0)		probabilities (Probabilities must sum to 1)
cat	**3.2**	exp →	**24.5**	normalize →	**0.13**
car	5.1		164.0		0.87
frog	-1.7		0.18		0.00

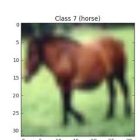
$L_i = -\log P(Y = y_i \mid X = x_i)$

→ L_i = -log(0.13) = **2.04**

Geometric interpretation



Wx



Array of 32x32x3