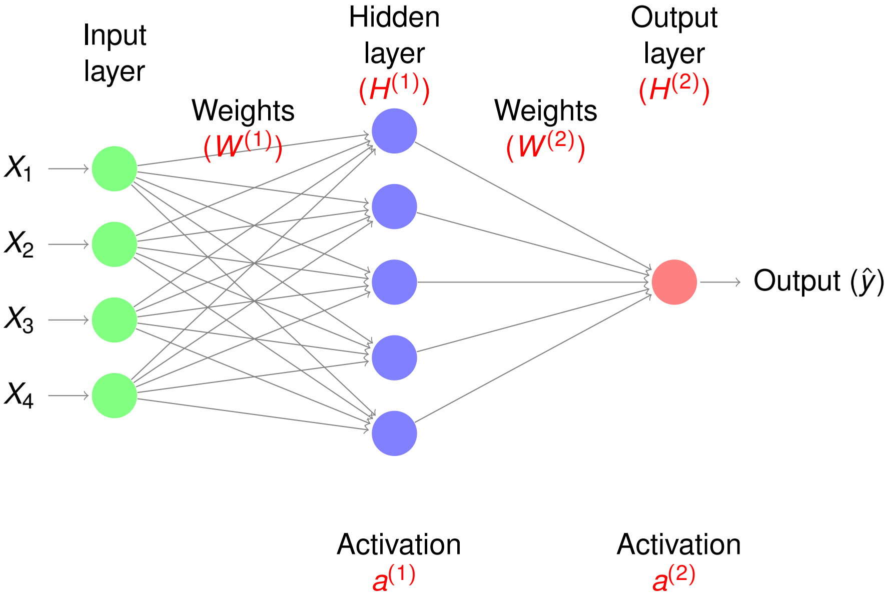
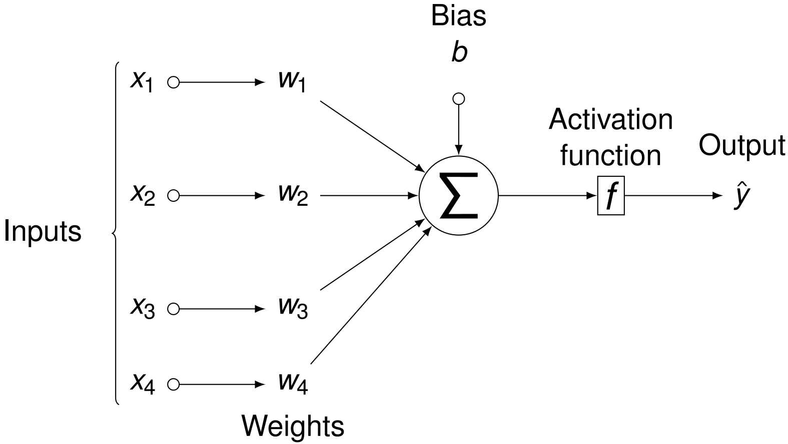
concordance=TRUE concordance=TRUE

Neural Networks

Steve

January 10, 2019


Input layer
Hidden layer
($H^{(1)}$)
Output layer
($H^{(2)}$)
Weights
($W^{(1)}$)
Weights
($W^{(2)}$)
X_1
X_2
X_3
X_4
Output ($\hat{y}$)
Activation
$a^{(1)}$
Activation
$a^{(2)}$


Bias
b
Inputs
x_1
x_2
x_3
x_4
w_1
w_2
w_3
w_4
Weights
$\sum$
Activation function
f
Output
$\hat{y}$

Components

- **Layers** - Input, Hidden and Output
  - **Neuron/Node/Unit** - Receives input and computes and output
- **Synapse** - Associated weights.
- **Activation function** - Introduces nonlinearity into the neuron output.
  - Sigmoid (Logistic Activation Function)
    $a(x) = \frac{1}{1+exp(-x)}$
  - Tanh (hyperbolic tangent Activation Function)
    $a(x) = tanh(x) = \frac{2}{1+exp(-2x)} - 1 = 2sigmoid(2x) - 1$
  - ReLU (Rectified Linear Unit Activation Function)
    $a(x) = max(0, x)$

```
##
## Attaching package:  'dplyr'
## The following objects are masked from
'package:stats':
##
##     filter, lag
## The following objects are masked from
'package:base':
##
##     intersect, setdiff, setequal, union
## -- Attaching packages
--------------------------------------- tidyverse
1.2.1 --
## v tibble  2.0.0     v purrr  0.2.5
## v readr  1.3.1     v stringr 1.3.1
## v tibble  2.0.0     v forcats 0.3.0
## -- Conflicts
------------------------------------------
tidyverse_conflicts() --
```

Feed-forward

Consider,

$$H_1^{(1)} = X_1 W_{1,1}^{(1)} + X_2 W_{2,1}^{(1)} + X_3 W_{3,1}^{(1)} + X_4 W_{4,1}^{(1)}$$

Feed-forward

Consider,

$$H_1^{(1)} = X_1 W_{1,1}^{(1)} + X_2 W_{2,1}^{(1)} + X_3 W_{3,1}^{(1)} + X_4 W_{4,1}^{(1)}$$

$$H_2^{(1)} = X_1 W_{1,2}^{(1)} + X_2 W_{2,2}^{(1)} + X_3 W_{3,2}^{(1)} + X_4 W_{4,2}^{(1)}$$

Feed-forward

Consider,

$$H_1^{(1)} = X_1 W_{1,1}^{(1)} + X_2 W_{2,1}^{(1)} + X_3 W_{3,1}^{(1)} + X_4 W_{4,1}^{(1)}$$

$$H_2^{(1)} = X_1 W_{1,2}^{(1)} + X_2 W_{2,2}^{(1)} + X_3 W_{3,2}^{(1)} + X_4 W_{4,2}^{(1)}$$

$$\vdots$$

Feed-forward

Consider,

$$H_1^{(1)} = X_1 W_{1,1}^{(1)} + X_2 W_{2,1}^{(1)} + X_3 W_{3,1}^{(1)} + X_4 W_{4,1}^{(1)}$$

$$H_2^{(1)} = X_1 W_{1,2}^{(1)} + X_2 W_{2,2}^{(1)} + X_3 W_{3,2}^{(1)} + X_4 W_{4,2}^{(1)}$$

$$\vdots$$

H^2 'component' is the sum of weighted inputs to each neuron.

$$H^{(1)} = XW^{(1)} \tag{1}$$

Feed-forward

Apply activation function to 1

$$a^{(1)} = a(H^{(1)}) \tag{2}$$

Propagate 2 to the output layer

$$H^{(2)} = a^{(1)} W^{(2)} \tag{3}$$

$$\implies \hat{y} = a^{(2)} = a(H^{(2)}) \tag{4}$$

Back propagation

- Estimate weights that ensures the model fits the training data well.

$$J = \sum \frac{1}{2}(y - \hat{y})^2 \tag{5}$$

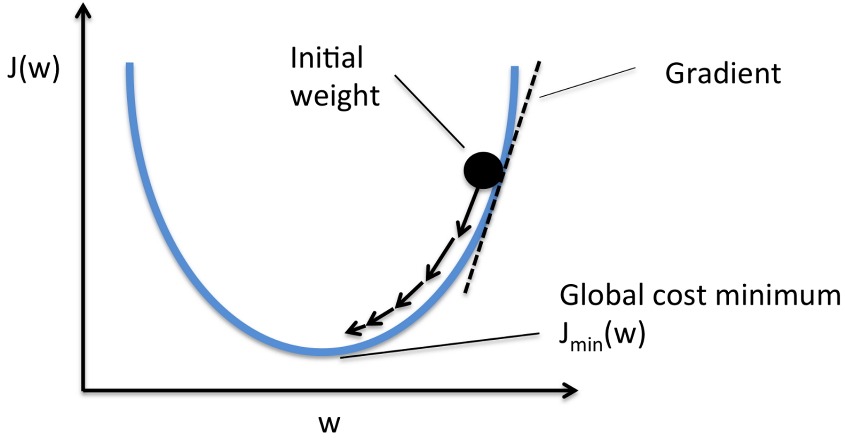
Back propagation

- Estimate weights that ensures the model fits the training data well.

$$J = \sum \frac{1}{2}(y - \hat{y})^2 \tag{5}$$

$$J(W) = \frac{1}{2}\sum \left(y - a(a(XW^{(1)})W^{(2)})\right)^2 \tag{6}$$

Gradient descent



Gradient descent

$$W_{t+1} = W_t - \gamma \Delta J(W_t)$$

Compute $\frac{\partial J}{\partial W^{(1)}}$ and $\frac{\partial J}{\partial W^{(2)}}$

$$\begin{aligned}
\frac{\partial J}{\partial W^{(2)}} &\approx -(y - \hat{y}) \frac{\partial \hat{y}}{\partial W^{(2)}} \\
&= -(y - \hat{y}) \frac{\partial \hat{y}}{\partial H^{(2)}} \frac{\partial H^{(2)}}{\partial W^{(2)}} \\
&= -(y - \hat{y}) a'(H^{(2)}) \frac{\partial H^{(2)}}{\partial W^{(2)}} \\
&= -(y - \hat{y}) a'(H^{(2)}) \frac{\partial H^{(2)}}{\partial W^{(2)}} \\
\implies \frac{\partial J}{\partial W^{(2)}} &= -a^{(1)T}(y - \hat{y}) a'(H^{(2)})
\end{aligned}$$

Gradient descent

$$
\begin{aligned}
\frac{\partial J}{\partial W^{(1)}} &\approx -(y-\hat{y})\frac{\partial \hat{y}}{\partial W^{(1)}} \\
&= -(y-\hat{y})\frac{\partial \hat{y}}{\partial H^{(2)}}\frac{\partial H^{(2)}}{\partial W^{(1)}} \\
&= -(y-\hat{y})a'(H^{(2)})\frac{\partial H^{(2)}}{\partial W^{(1)}} \\
&= -(y-\hat{y})a'(H^{(2)})\frac{\partial H^{(2)}}{\partial a^{(1)}}\frac{\partial a^{(1)}}{\partial W^{(1)}} \\
&= -(y-\hat{y})a'(H^{(2)})W^{(2)T}\frac{\partial a^{(1)}}{\partial W^{(1)}} \\
&= -(y-\hat{y})a'(H^{(2)})W^{(2)T}\frac{\partial a^{(2)}}{\partial H^{(2)}}\frac{\partial H^{(2)}}{\partial W^{(1)}} \\
\Longrightarrow \frac{\partial J}{\partial W^{(1)}} &= -X^T(y-\hat{y})a'(H^{(2)})W^{(2)T}a'(H^{(1)})
\end{aligned}
$$

Remarks

1. Starting values
2. Overfitting and Stopping Criterion
   - Reduce training error to some predetermined threshold $\longrightarrow$ overffiting.
   - Regularization

$$J(W) = \sum \frac{1}{2}(y - \hat{y})^2/n + \frac{1}{2}\lambda\left(\sum W_1 + \sum W_2\right)$$

λ - Regularization hyper parameter

Remarks

3. Convergence and Local Minima