Introduction to Neural Networks

Hunter Glanz

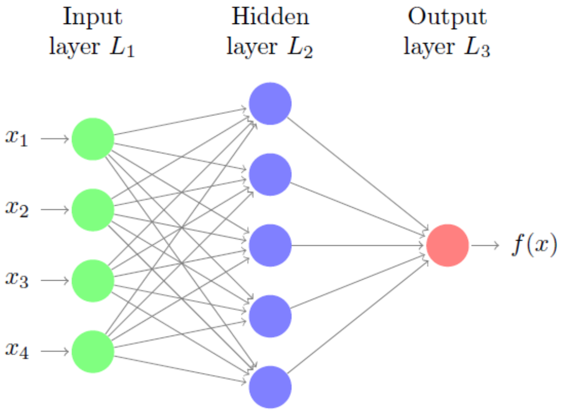
OUTLINE

Overview

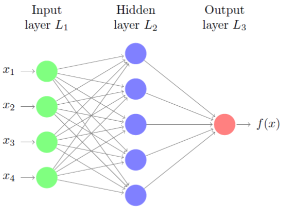
- Neural networks are highly parameterized models inspired by the human brain

Overview

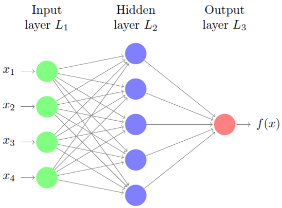
- Neural networks are highly parameterized models inspired by the human brain



More Detail

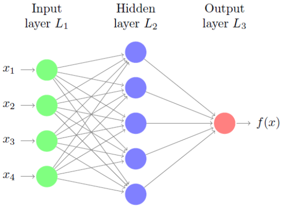


More Detail



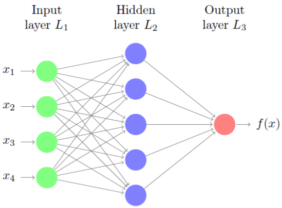
- Four predictors or inputs x_j

More Detail



- Four predictors or inputs x_j
- Five hidden units $a_\ell = g(w_{\ell 0}^{(1)} + \sum_{j=1}^{4} w_{\ell j}^{(1)} x_j)$

More Detail



- Four predictors or inputs x_j
- Five hidden units $a_\ell = g(w_{\ell 0}^{(1)} + \sum_{j=1}^{4} w_{\ell j}^{(1)} x_j)$
- Single output unit $o = h(w_0^{(2)} + \sum_{\ell=1}^{5} w_\ell^{(2)} a_\ell)$

So Many Weights!

- Four predictors or inputs x_j

So Many Weights!

- Four predictors or inputs x_j
- Five hidden units $a_\ell = g(w_{\ell 0}^{(1)} + \sum_{j=1}^{4} w_{\ell j}^{(1)} x_j)$

So Many Weights!

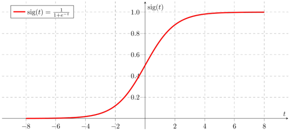
- Four predictors or inputs x_j
- Five hidden units $a_\ell = g(w_{\ell 0}^{(1)} + \sum_{j=1}^{4} w_{\ell j}^{(1)} x_j)$
- Single output unit $o = h(w_0^{(2)} + \sum_{\ell=1}^{5} w_\ell^{(2)} a_\ell)$

So Many Weights!

- Four predictors or inputs x_j
- Five hidden units $a_\ell = g(w_{\ell 0}^{(1)} + \sum_{j=1}^{4} w_{\ell j}^{(1)} x_j)$
- Single output unit $o = h(w_0^{(2)} + \sum_{\ell=1}^{5} w_\ell^{(2)} a_\ell)$
- Typically $g(t) = 1/(1 + e^{-t})$ – a sigmoid

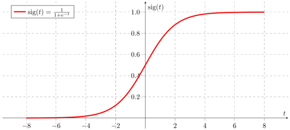
So Many Weights!

- Four predictors or inputs x_j
- Five hidden units $a_\ell = g(w_{\ell 0}^{(1)} + \sum_{j=1}^{4} w_{\ell j}^{(1)} x_j)$
- Single output unit $o = h(w_0^{(2)} + \sum_{\ell=1}^{5} w_\ell^{(2)} a_\ell)$
- Typically $g(t) = 1/(1 + e^{-t})$ – a sigmoid



So Many Weights!

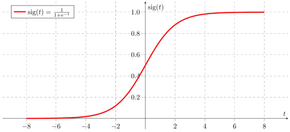
- Four predictors or inputs x_j
- Five hidden units $a_\ell = g(w_{\ell 0}^{(1)} + \sum_{j=1}^{4} w_{\ell j}^{(1)} x_j)$
- Single output unit $o = h(w_0^{(2)} + \sum_{\ell=1}^{5} w_\ell^{(2)} a_\ell)$
- Typically $g(t) = 1/(1 + e^{-t})$ – a sigmoid



- For quantitative regression, h is typically the identity
- For classification, h is once again the sigmoid

So Many Weights!

- Four predictors or inputs x_j
- Five hidden units $a_\ell = g(w_{\ell 0}^{(1)} + \sum_{j=1}^{4} w_{\ell j}^{(1)} x_j)$
- Single output unit $o = h(w_0^{(2)} + \sum_{\ell=1}^{5} w_\ell^{(2)} a_\ell)$
- Typically $g(t) = 1/(1 + e^{-t})$ – a sigmoid



- For quantitative regression, h is typically the identity
- For classification, h is once again the sigmoid

How are neural networks different from the ensemble methods we've discussed this term?

Fitting a Neural Network

- **Iteratively fit the weights** to minimize some loss function, L
  - L depends on your problem, but think of RSS or classification error

Fitting a Neural Network

- **Iteratively fit the weights** to minimize some loss function, L
  - L depends on your problem, but think of RSS or classification error
- In fact...we want to minimize:

$$\text{minimize}_{\mathcal{W}} \quad \frac{1}{n}\sum_{i=1}^{n} L[y_i, f(x_i; \mathcal{W})] + \lambda J(\mathcal{W}) \tag{1.1}$$

where $\mathcal{W}$ is the collection of weights, J is a nonnegative regularization term, and $\lambda \geq 0$ is a tuning parameter

Tuning Parameters

- Number of hidden layers
- Number of units in each hidden layer
- Starting values for the weights
- Choice of non-linearities (the form of the g function)
- Choice of regularization
- Choice of stopping time

Tuning Parameters

- Number of hidden layers
- Number of units in each hidden layer
- Starting values for the weights
- Choice of non-linearities (the form of the g function)
- Choice of regularization
- Choice of stopping time
- **How do we determine these?!**

Tuning Parameters

- Number of hidden layers
- Number of units in each hidden layer
- Starting values for the weights
- Choice of non-linearities (the form of the g function)
- Choice of regularization
- Choice of stopping time
- **How do we determine these?!**

A little bit of cross-validation

Tuning Parameter Details

- Too many weights can lead to overfitting and a global minimum of the loss

Tuning Parameter Details

- Too many weights can lead to overfitting and a global minimum of the loss
- Current wisdom:

Tuning Parameter Details

- Too many weights can lead to overfitting and a global minimum of the loss
- Current wisdom:
  - Better to have large number of hidden units (5 to 100)

Tuning Parameter Details

- Too many weights can lead to overfitting and a global minimum of the loss
- Current wisdom:
  - Better to have large number of hidden units (5 to 100)
  - Control model complexity with regularization (**cross-validation**)
    - Regularization slows the rate of overfitting

Tuning Parameter Details

- Too many weights can lead to overfitting and a global minimum of the loss
- Current wisdom:
  - Better to have large number of hidden units (5 to 100)
  - Control model complexity with regularization (**cross-validation**)
    - Regularization slows the rate of overfitting
  - More hidden layers increases complexity, but tends to be task specific

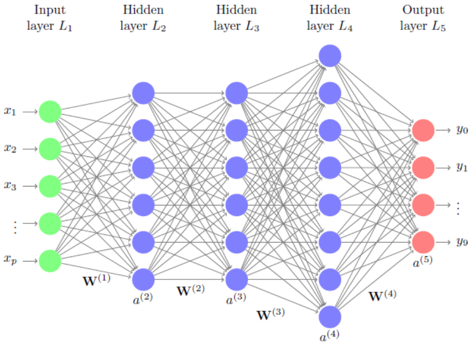
Tuning Parameter Details

- Too many weights can lead to overfitting and a global minimum of the loss
- Current wisdom:
  - Better to have large number of hidden units (5 to 100)
  - Control model complexity with regularization (**cross-validation**)
    - Regularization slows the rate of overfitting
  - More hidden layers increases complexity, but tends to be task specific
  - Stopping time can also be determined with **cross-validation**, but is less of a problem with regularization

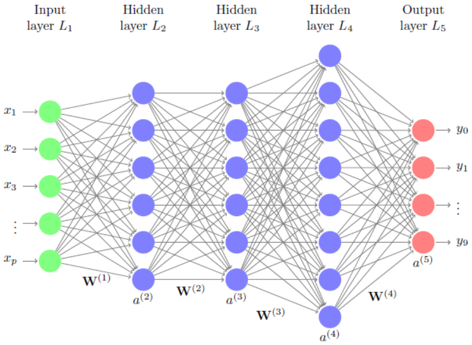
Tuning Parameter Details

- Too many weights can lead to overfitting and a global minimum of the loss
- Current wisdom:
  - Better to have large number of hidden units (5 to 100)
  - Control model complexity with regularization (**cross-validation**)
    - Regularization slows the rate of overfitting
  - More hidden layers increases complexity, but tends to be task specific
  - Stopping time can also be determined with **cross-validation**, but is less of a problem with regularization
  - Choice of starting weight values can affect movement of the algorithm and where it ends up
    - Shouldn't be zero
    - Shouldn't be large
    - Try multiple sets

More Complex

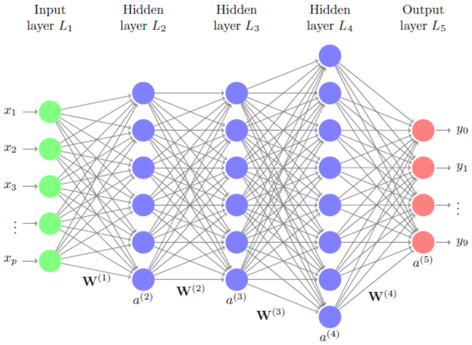


More Complex



- Non-linear extension of traditional linear models

More Complex



- Non-linear extension of traditional linear models
- From very regularized and constrained, to very flexible