Neural network classification and Regression trees

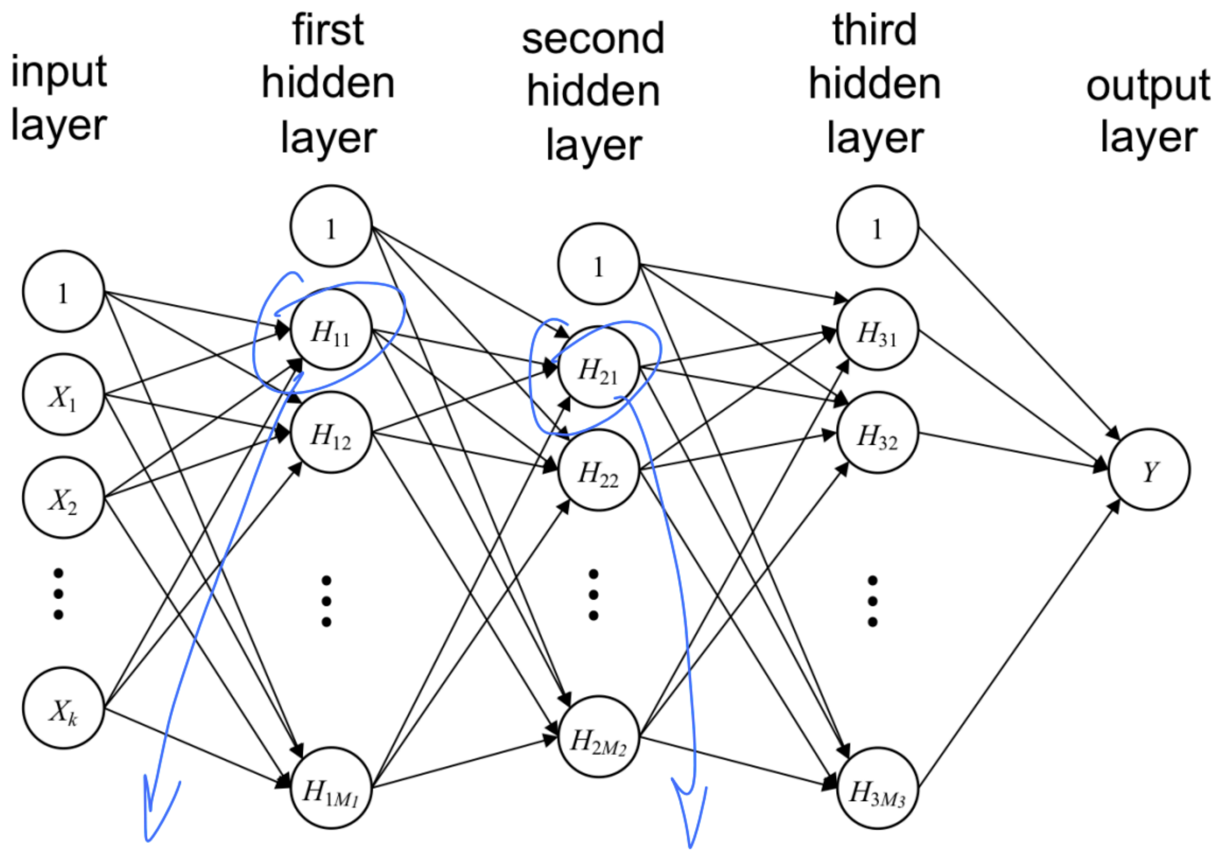
More about neural networks

1. Think how each node is a function of all input predictors;

2. The activation function works as nonlinear transformation;

3. 1 & 2 together can approximate complex relationships among predictors, think how you need to specify components in regression;

4. You should be able to do manual calculations if given weights and activation functions in neural networks.



$H_{11} = \sigma(f_{11}(1, X_1, X_2, \cdots X_k))$

$\vdots$

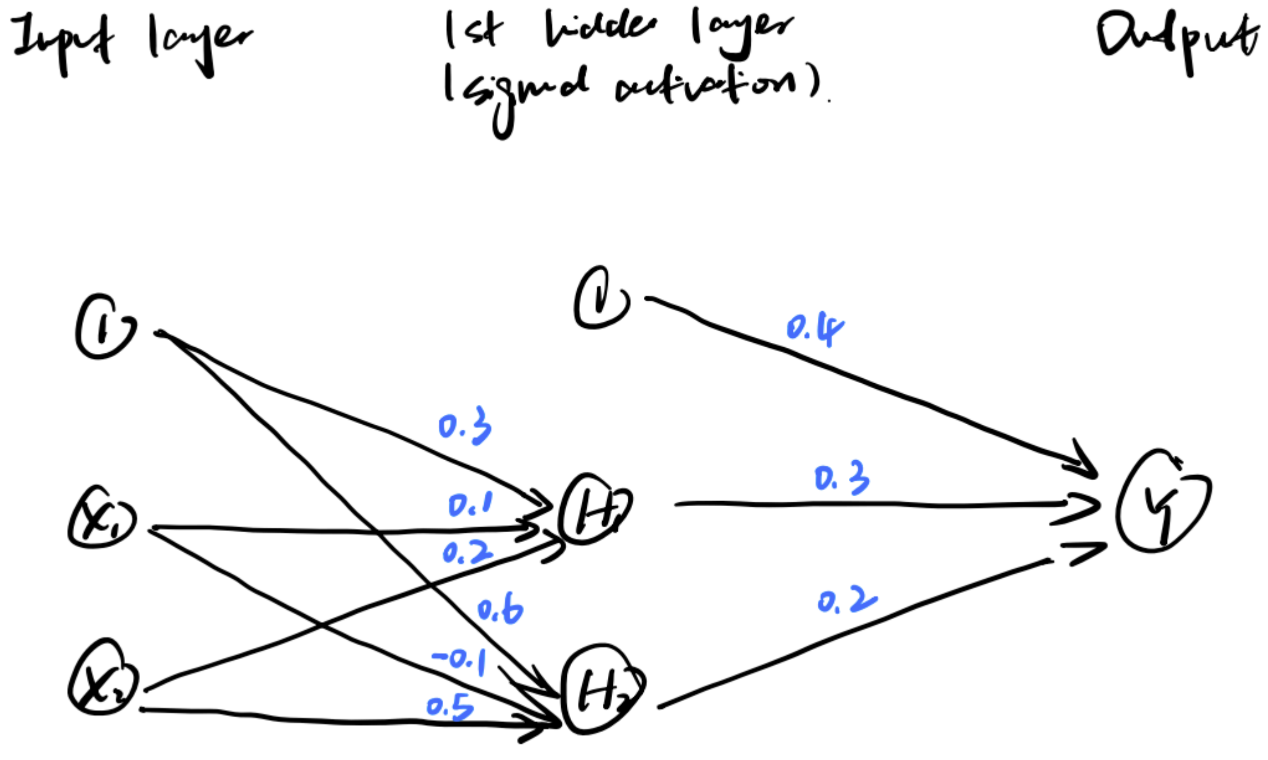
$H_{1M_1} = \sigma(f_{1M_1}(1, X_1, \cdots X_k))$

$\sigma(\cdot)$: sigmoid function.

$H_{21} = \sigma(f_{21}(1, H_{11}, \cdots H_{2M_2}))$

$\vdots$

Easy example of nnet calculation



Given x_1, x_2.

$$H_1 = \frac{\exp\{0.3 + 0.1x_1 + 0.2x_2\}}{1 + \exp\{0.3 + 0.1x_1 + 0.2x_2\}}$$

$$\hat{y} = 0.4 + 0.3 \cdot H_1 + 0.2 H_2.$$

$$H_2 = \frac{\exp\{0.6 - 0.1x_1 + 0.5x_2\}}{1 + \exp\{0.6 - 0.1x_1 + 0.5x_2\}}$$

Weakest link pruning

- Goal: find the unique smallest subtree $T_\alpha \subseteq T_0$ that minimize the cost complexity criterion;
- Each time, collapse one internal node that produces the smallest per-node increase in impurity, and continue this procedure until getting the single-node tree. This sequence of subtrees is guarantee to have T_α;
- T_α is associated with T_0 and α.

Feature importance

- Visual inspection: how many times the predictors appears in internal nodes, do they appear close the root, do their branches have long length;

- Numerical measure: for some x_i, sum the reductions in impurity measure for internal nodes when the split is based on this x_i.

Difference among impurity measures

Gini index and cross-entropy loss are more sensitive to changes in node probabilities than the misclassification rate (example in ESL P.311):

Class 1	Class 2	Predict Class 1		Predict Class 2		Misclassification rate	Gini index	Cross-entropy
		300	100	100	300	200/800	400*2*(300/400)*(100/400)+400*2*(300/400)*(100/400)	-2*400*0.75*log(0.75)-2*400*0.25*log(0.25)
400	400							
		400	200	0	200	200/800	600*2*(400/600)*(200/600)	-600*0.667*log(0.667)-600*0.333*log(0.333)