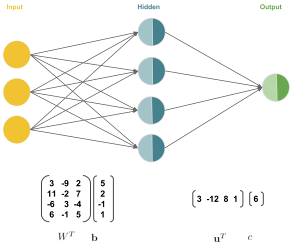
Introduction to Machine Learning

Neural Networks
MLP: Matrix Notation



Learning goals

- Compact representation of neural network equations
- Vector notation for neuron layers
- Vector and matrix notation of bias and weight parameters

SINGLE HIDDEN LAYER NETWORKS: INPUT NOTATION

- The input $\mathbf{x}$ is a column vector with dimensions $p \times 1$.
- $\mathbf{W}$ is a weight matrix with dimensions $p \times m$, where m is the amount of hidden neurons:

$$\mathbf{W} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{p,1} & w_{p,2} & \cdots & w_{p,m} \end{pmatrix}$$

SINGLE HIDDEN LAYER NETWORKS: HIDDEN LAYER NOTATION

- For example, to obtain z_1, we pick the first column of W:

$$\mathbf{W}_1 = \begin{pmatrix} w_{1,1} \\ w_{2,1} \\ \vdots \\ w_{p,1} \end{pmatrix}$$

  and compute

$$z_1 = \sigma(\mathbf{W}_1^T \mathbf{x} + b_1) \ ,$$

  where b_1 is the bias of the first hidden neuron and $\sigma : \mathbb{R} \to \mathbb{R}$ is an activation function.

SINGLE HIDDEN LAYER NETWORKS: HIDDEN LAYER NOTATION

- The network has m hidden neurons $z_1, \ldots, z_m$ with

$$z_j = \sigma(\mathbf{W}_j^T \mathbf{x} + b_j), j \in \{1, \ldots, m\}$$

    - $z_{j,in} = \mathbf{W}_j^T \mathbf{x} + b_j$
    - $z_{j,out} = \sigma(z_{j,in}) = \sigma(\mathbf{W}_j^T \mathbf{x} + b_j)$
- $\mathbf{z}_{in} = (z_{1,in}, \ldots, z_{m,in})^T = \mathbf{W}^T \mathbf{x} + \mathbf{b}$
  (Note: $\mathbf{W}^T \mathbf{x} = (\mathbf{x}^T \mathbf{W})^T$)
- $\mathbf{z} = \mathbf{z}_{out} = \sigma(\mathbf{z}_{in}) = \sigma(\mathbf{W}^T \mathbf{x} + \mathbf{b})$, where the (hidden layer) activation function σ is applied element-wise to $\mathbf{z}_{in}$.

SINGLE HIDDEN LAYER NETWORKS: BIAS TERM NOTATION

- We sometimes omit the bias term by adding a constant feature to the input $\tilde{\mathbf{x}} = (1, x_1, ..., x_p)$ and by adding the bias term to the weight matrix
$$\tilde{\mathbf{W}} = (\mathbf{b}, \mathbf{W}_1, ..., \mathbf{W}_p).$$

- **Note**: For simplification purposes, we will not explicitly represent the bias term graphically in the following. However, the above "trick" makes it straightforward to represent it graphically.

SINGLE HIDDEN LAYER NETWORKS: OUTPUT LAYER NOTATION

- For regression or binary classification: one output unit f where
  - $f_{in} = \mathbf{u}^T \mathbf{z} + c$, i.e. a linear combination of derived features plus the bias term c of the output neuron, and
  - $f(\mathbf{x}) = f_{out} = \tau(f_{in}) = \tau(\mathbf{u}^T \mathbf{z} + c)$, where τ is the output activation function.
- For regression τ is the identity function.
- For binary classification, τ is a sigmoid function.
- **Note**: The purpose of the hidden-layer activation function σ is to introduce non-linearities so that the network is able to learn complex functions whereas the purpose of τ is merely to get the final score to the same range as the target.

SINGLE HIDDEN LAYER NETWORKS: MULTIPLE INPUTS NOTATION

- It is possible to feed multiple inputs to a neural network simultaneously.
- The inputs $\mathbf{x}^{(i)}$, for $i \in \{1, \ldots, n\}$, are arranged as rows in the **design matrix X**.
  - $\mathbf{X}$ is a $(n \times p)$-matrix.
- The weighted sum in the hidden layer is now computed as $\mathbf{XW} + \boldsymbol{B}$, where,
  - $\mathbf{W}$, as usual, is a $(p \times m)$ matrix, and,
  - $\boldsymbol{B}$ is a $(n \times m)$ matrix containing the bias vector $\mathbf{b}$ (duplicated) as the rows of the matrix.

SINGLE HIDDEN LAYER NETWORKS: MULTIPLE INPUTS NOTATION

- The *matrix* of hidden activations $\boldsymbol{Z} = \sigma(\mathbf{XW} + \boldsymbol{B})$
  - $\boldsymbol{Z}$ is a $(n \times m)$ matrix.

- The final output of the network, which contains a prediction for each input, is $\tau(\boldsymbol{Z}\mathbf{u} + \boldsymbol{C})$, where
  - $\mathbf{u}$ is the vector of weights of the output neuron, and,
  - $\boldsymbol{C}$ is a $(n \times 1)$ matrix whose elements are the (scalar) bias c of the output neuron.