Applied Machine Learning

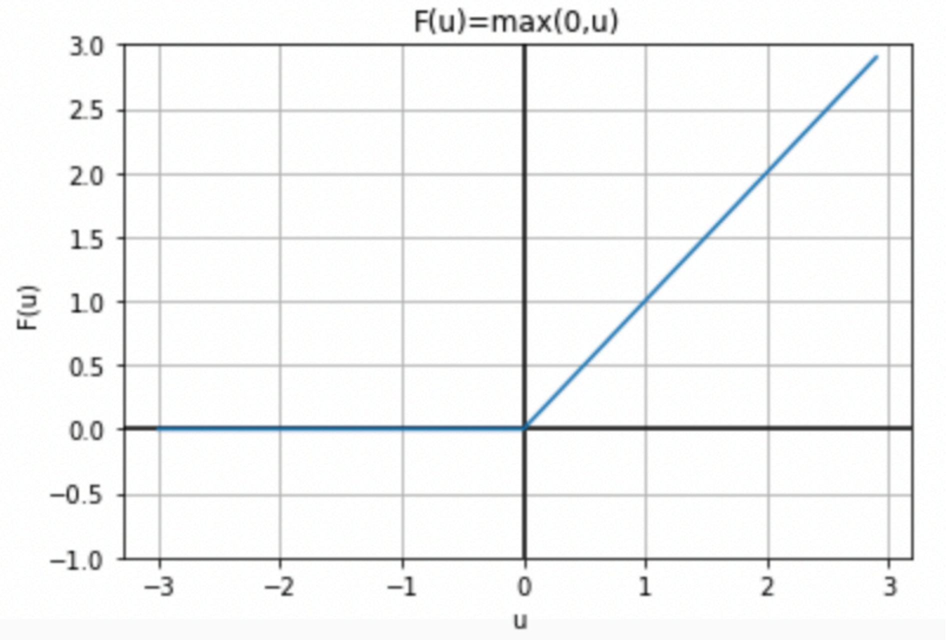
Simple Neural Network Classifier

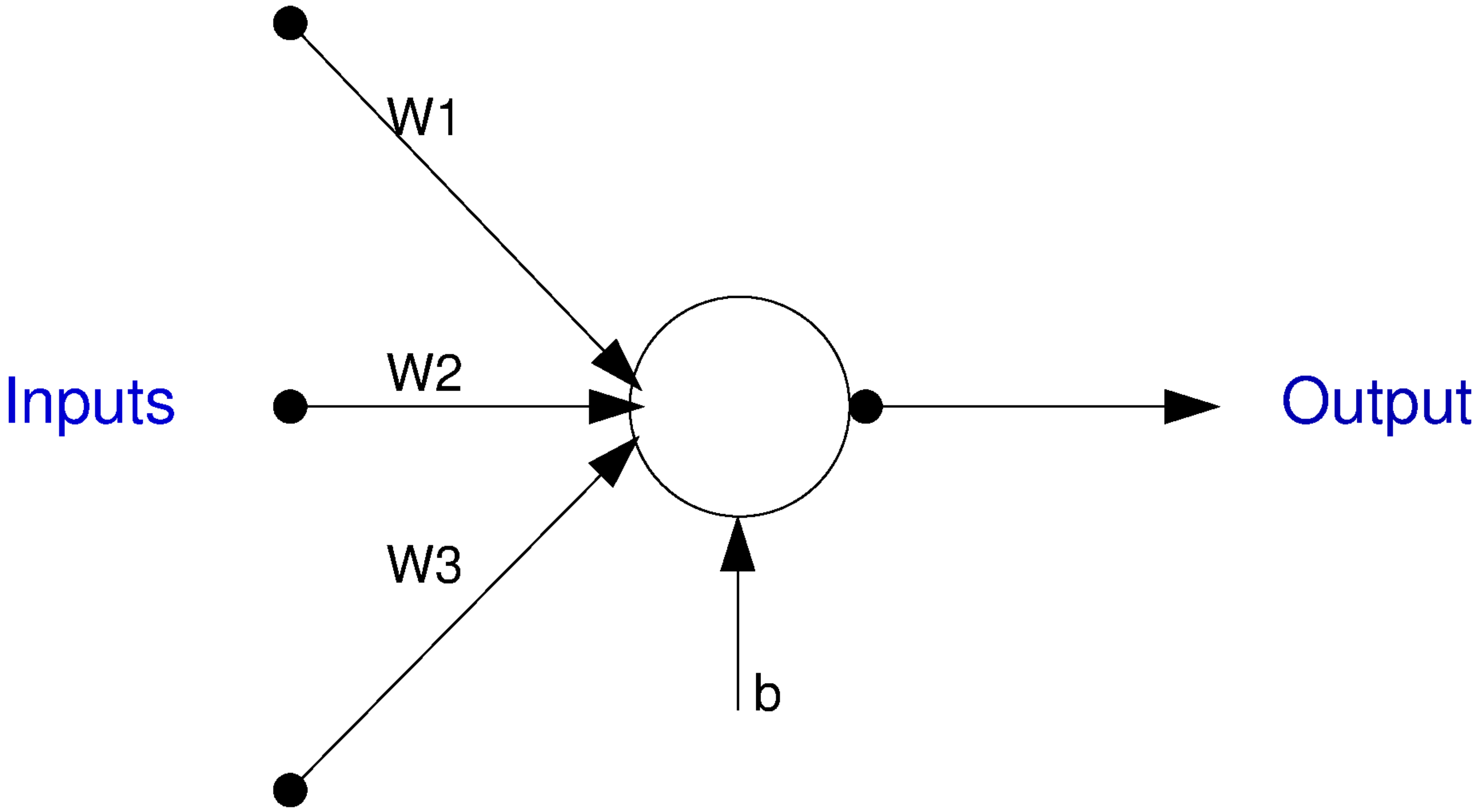
Simple Neural Network Classifier

- Structure of a Unit in a Neural Network
- Example of activation function: Rectified Linear Unit - ReLU
- Simple Neural Network classifier
  - ReLU and Softmax layers
- Classification process

Neural Network Unit

- One unit
- Input: vector $\mathbf{x}$
- Parameters:
  - Weights: vector $\mathbf{w}$
  - Bias: scalar b
- Associated function:
  - Linear part: $u = \sum_i w_i x_i + b$. Vectorized: $u = \mathbf{w}^\top \mathbf{x} + b$
  - Activation:
    - nonlinear Rectified Linear Unit - **ReLU:** $F(u) = \max(0, u)$
    - 0: point at one side of plane
    - u: point at other side of plane
      - grows with distance from plane

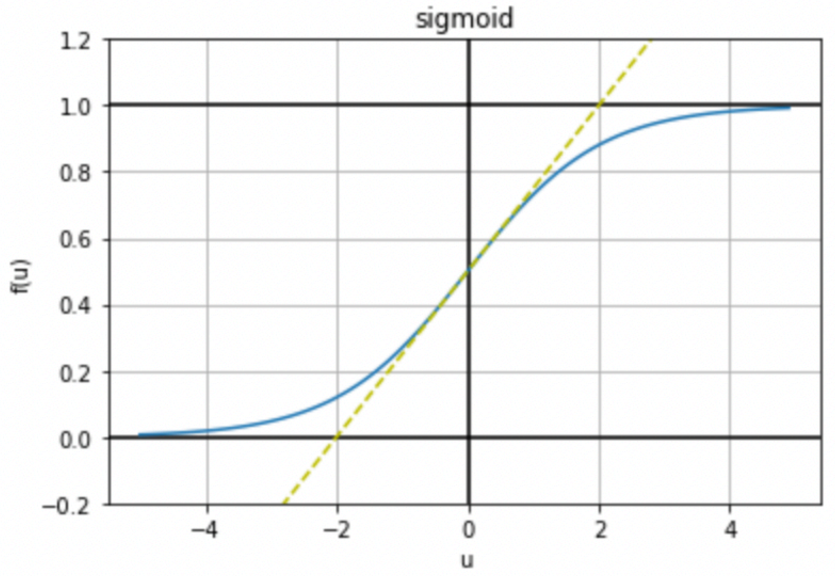




Simple Neural Network Classifier

- One unit per class
  - output: probability that input features correspond to class
- Classifier for C classes
  - outputs: $\mathbf{o} = \begin{bmatrix} o_1 \\ \vdots \\ o_C \end{bmatrix}$, o_j: output of unit j
  - Parameters of unit j
    - weights: vector $\mathbf{w}^{(j)} = \begin{bmatrix} w_1^{(j)} \\ \vdots \\ w_n^{(j)} \end{bmatrix}$, bias: scalar $b^{(j)}$

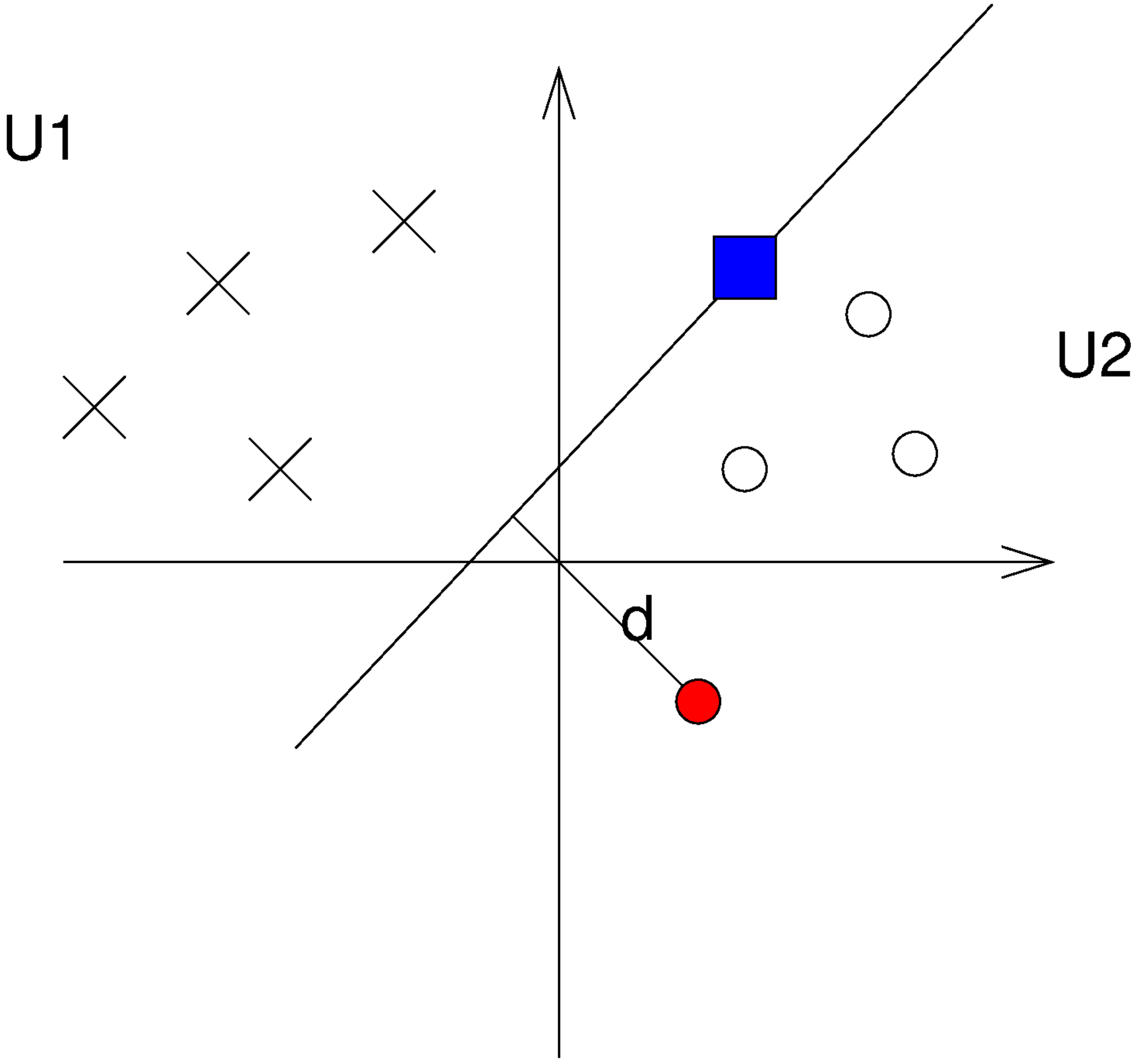
- Goal: output of unit j models probability of input $\mathbf{x}_i$ to correspond to class j
- Softmax function for vector with C class outputs $\mathbf{o}$
  - softmax$(u) = \mathbf{s}(\mathbf{o}) = \frac{1}{\sum_k e^{o_k}} \begin{bmatrix} e^{o_1} \\ \vdots \\ e^{o_C} \end{bmatrix}$



  - Class for data item $\mathbf{x}_i$:
    - one-hot vector: $\mathbf{y}_i = \begin{bmatrix} y_0 \\ \vdots \\ y_C \end{bmatrix} y_j = 1, y_{i \neq j} = 0$
  - Probability of input $\mathbf{x}_i$ corresponding to class j
    - $p(y_j = 1 \,|\, \mathbf{x}_i, \mathbf{w}^{(j)}, b^{(j)}) = s_j(\mathbf{o}(\mathbf{x}_i, \mathbf{w}^{(j)}, b^{(j)}))$

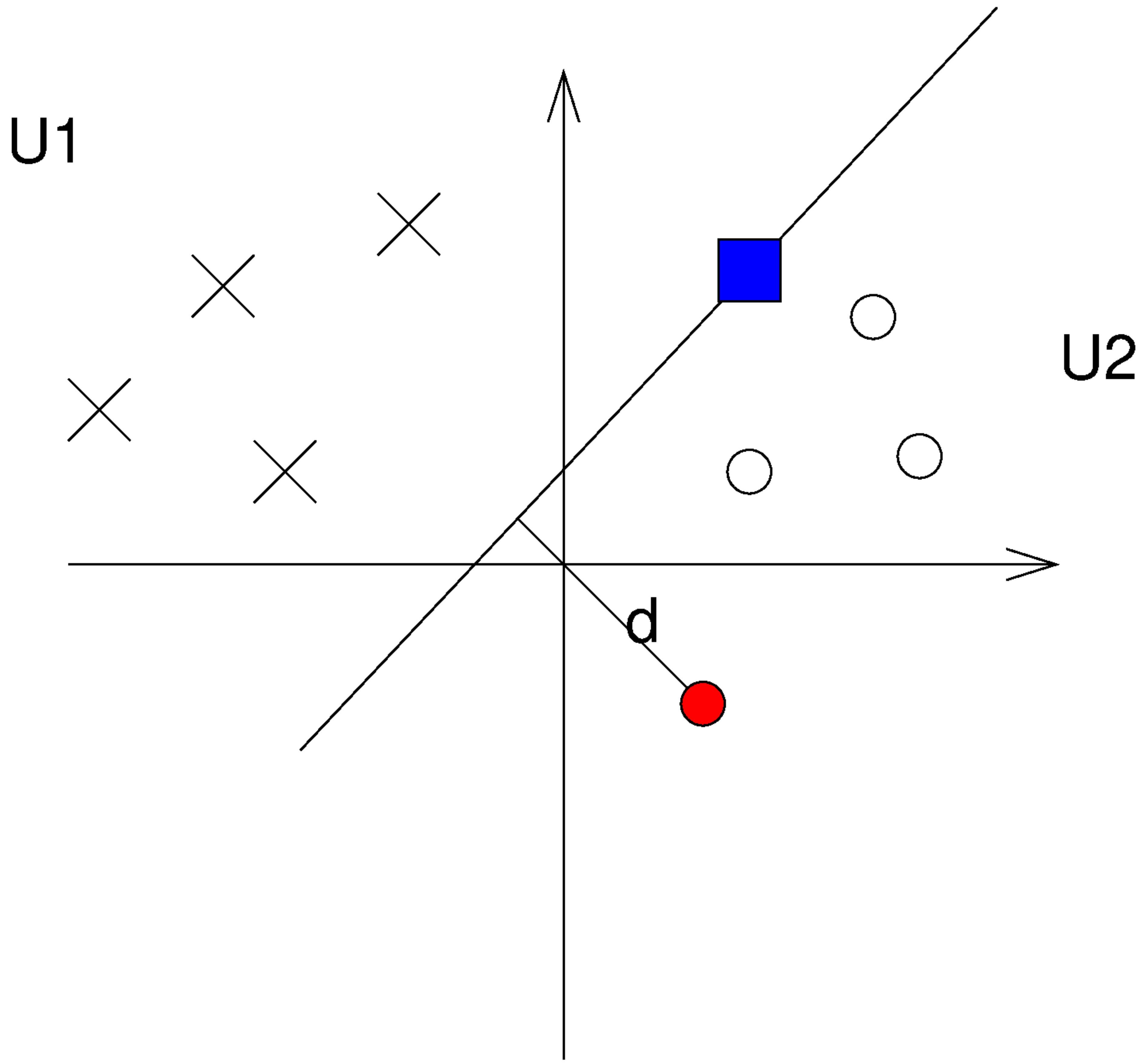
Simple Neural Network Classifier

- Classifying among two classes: × , O
  - ReLU units: u_1, u_2
    - $\times : \begin{cases} u_1 & \text{grows with distance to plane} \\ u_2 & 0 \end{cases}$
    - $\mathrm{O} : \begin{cases} u_1 & 0 \\ u_2 & \text{grows with distance to plane} \end{cases}$
    - $\square : \begin{cases} u_1 = 0 & \Rightarrow s_1 = \dfrac{e^0}{e^0 + e^0} = \dfrac{1}{2} \\ u_2 = 0 & \Rightarrow s_2 = \dfrac{e^0}{e^0 + e^0} = \dfrac{1}{2} \end{cases}$



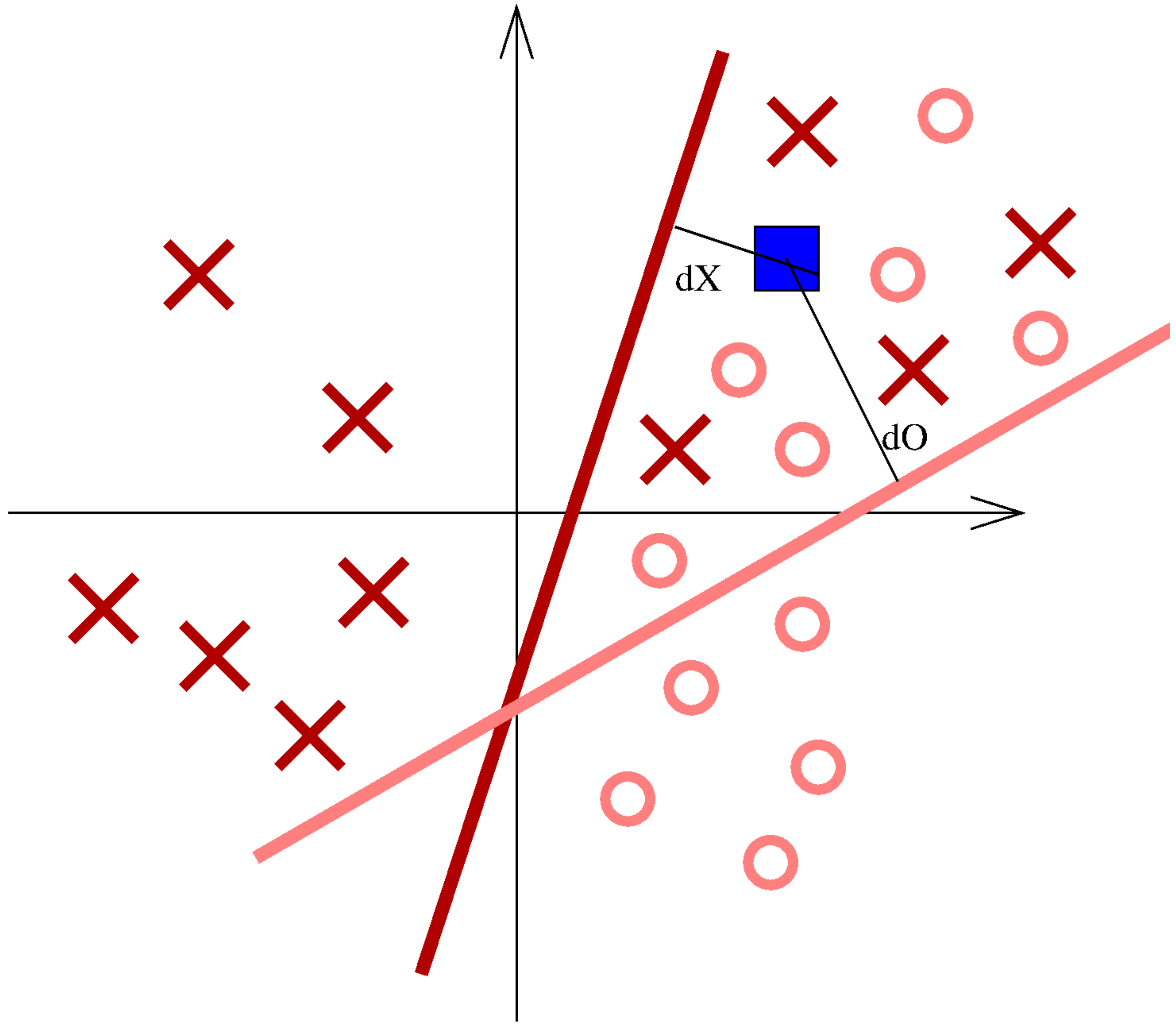
Simple Neural Network Classifier

- Classifying among two classes: × , O
  - ReLU units: u_1, u_2
    - $\times : \begin{cases} u_1 & \text{grows with distance to plane} \\ u_2 & 0 \end{cases}$
    - $\mathbf{o} : \begin{cases} u_1 = 0 & \Rightarrow s_1 = \dfrac{e^0}{e^0 + e^\delta} = \dfrac{1}{1 + e^\delta} \\ u_2 = \delta & \Rightarrow s_2 = \dfrac{e^\delta}{e^0 + e^\delta} = \dfrac{e^\delta}{1 + e^\delta} \end{cases}$
    - $\square : \begin{cases} u_1 = 0 & \Rightarrow s_1 = \dfrac{e^0}{e^0 + e^0} = \dfrac{1}{2} \\ u_2 = 0 & \Rightarrow s_2 = \dfrac{e^0}{e^0 + e^0} = \dfrac{1}{2} \end{cases}$



Classification Regions

- Regions
  - only ×: positive for u_1
  - only O: positive for u_2
  - both × and O: positive for both u_1 and u_2
    - split in two parts
      - higher probability for u_1: ×
      - higher probability for u_2: O
  - no dataset item in region, probabilities: $\frac{1}{2}$



Simple Neural Network Classifier

- Structure of a Unit in a Neural Network
- Example of activation function: Rectified Linear Unit - ReLU
- Simple Neural Network classifier
    - ReLU and Softmax layers
- Classification process

Applied Machine Learning

Simple Neural Network Classifier