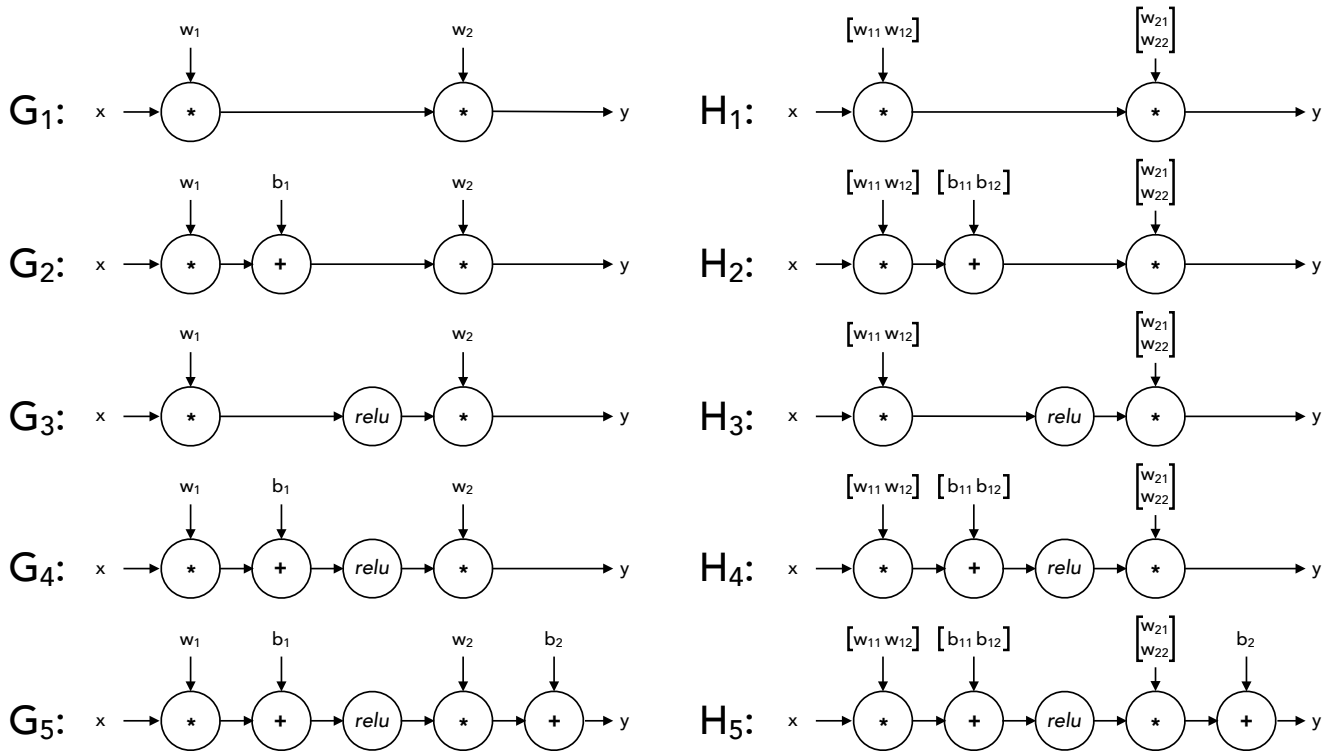# Q1. Machine Learning: Potpourri

(a) What it the **minimum** number of parameters needed to fully model a joint distribution $P(Y, F_1, F_2, ..., F_n)$ over label Y and n features F_i? Assume binary class where each feature can possibly take on k distinct values.

(b) Under the **Naive Bayes assumption**, what is the **minimum** number of parameters needed to model a joint distribution $P(Y, F_1, F_2, ..., F_n)$ over label Y and n features F_i? Assume binary class where each feature can take on k distinct values.

(c) You suspect that you are overfitting with your Naive Bayes with Laplace Smoothing. How would you adjust the strength k in Laplace Smoothing?

     ○ Increase k ○ Decrease k

(d) While using Naive Bayes with Laplace Smoothing, increasing the strength k in Laplace Smoothing can:

     ☐ Increase training error   ☐ Decrease training error
     ☐ Increase validation error   ☐ Decrease validation error

(e) It is possible for the perceptron algorithm to never terminate on a dataset that is linearly separable in its feature space.

     ○ True   ○ False

(f) If the perceptron algorithm terminates, then it is guaranteed to find a max-margin separating decision boundary.

     ○ True   ○ False

(g) In binary perceptron where the initial weight vector is $\vec{0}$, the final weight vector can be written as a linear combination of the training data feature vectors.

     ○ True   ○ False

(h) For binary class classification, logistic regression produces a linear decision boundary.

     ○ True   ○ False

(i) In the binary classification case, logistic regression is exactly equivalent to a single-layer neural network with a sigmoid activation and the cross-entropy loss function.

     ○ True   ○ False

(j) You train a linear classifier on 1,000 training points and discover that the training accuracy is only 50%. Which of the following, if done in isolation, has a good chance of improving your training accuracy?

     ☐ Add novel features   ☐ Train on more data

(k) You now try training a neural network but you find that the training accuracy is still very low. Which of the following, if done in isolation, has a good chance of improving your training accuracy?

     ☐ Add more hidden layers   ☐ Add more units to the hidden layers

Q2. Neural Networks: Representation

G_1: with weights w_1, w_2 — $x \to (*) \to (*) \to y$

H_1: with weights $[w_{11}\ w_{12}]$, $\begin{bmatrix} w_{21} \\ w_{22} \end{bmatrix}$ — $x \to (*) \to (*) \to y$

G_2: with w_1, b_1, w_2 — $x \to (*) \to (+) \to (*) \to y$

H_2: with $[w_{11}\ w_{12}]$, $[b_{11}\ b_{12}]$, $\begin{bmatrix} w_{21} \\ w_{22} \end{bmatrix}$ — $x \to (*) \to (+) \to (*) \to y$

G_3: with w_1, w_2 — $x \to (*) \to relu \to (*) \to y$

H_3: with $[w_{11}\ w_{12}]$, $\begin{bmatrix} w_{21} \\ w_{22} \end{bmatrix}$ — $x \to (*) \to relu \to (*) \to y$

G_4: with w_1, b_1, w_2 — $x \to (*) \to (+) \to relu \to (*) \to y$

H_4: with $[w_{11}\ w_{12}]$, $[b_{11}\ b_{12}]$, $\begin{bmatrix} w_{21} \\ w_{22} \end{bmatrix}$ — $x \to (*) \to (+) \to relu \to (*) \to y$

G_5: with w_1, b_1, w_2, b_2 — $x \to (*) \to (+) \to relu \to (*) \to (+) \to y$

H_5: with $[w_{11}\ w_{12}]$, $[b_{11}\ b_{12}]$, $\begin{bmatrix} w_{21} \\ w_{22} \end{bmatrix}$, b_2 — $x \to (*) \to (+) \to relu \to (*) \to (+) \to y$

For each of the piecewise-linear functions below, mark all networks from the list above that can represent the function **exactly** on the range $x \in (-\infty, \infty)$. In the networks above, *relu* denotes the element-wise ReLU nonlinearity: $relu(z) = max(0, z)$. The networks G_i use 1-dimensional layers, while the networks H_i have some 2-dimensional intermediate layers.

(a) [plot: increasing straight line through origin]

- ☐ G_1 ☐ H_1
- ☐ G_2 ☐ H_2
- ☐ G_3 ☐ H_3
- ☐ G_4 ☐ H_4
- ☐ G_5 ☐ H_5
- ◯ None of the above

[plot: decreasing straight line]

- ☐ G_1 ☐ H_1
- ☐ G_2 ☐ H_2
- ☐ G_3 ☐ H_3
- ☐ G_4 ☐ H_4
- ☐ G_5 ☐ H_5
- ◯ None of the above

(b) [plot: piecewise-linear increasing with kink near origin]

- ☐ G_1 ☐ H_1
- ☐ G_2 ☐ H_2
- ☐ G_3 ☐ H_3
- ☐ G_4 ☐ H_4
- ☐ G_5 ☐ H_5
- ◯ None of the above

[plot: V-shaped function with flat ends at y=2]

- ☐ G_1 ☐ H_1
- ☐ G_2 ☐ H_2
- ☐ G_3 ☐ H_3
- ☐ G_4 ☐ H_4
- ☐ G_5 ☐ H_5
- ◯ None of the above

(c) [plot: flat at y=1, decreasing, then flat at y=-1]

- ☐ G_1 ☐ H_1
- ☐ G_2 ☐ H_2
- ☐ G_3 ☐ H_3
- ☐ G_4 ☐ H_4
- ☐ G_5 ☐ H_5
- ◯ None of the above

[plot: inverted-V / peak shape]

- ☐ G_1 ☐ H_1
- ☐ G_2 ☐ H_2
- ☐ G_3 ☐ H_3
- ☐ G_4 ☐ H_4
- ☐ G_5 ☐ H_5
- ◯ None of the above