Tecnológico de Monterrey

Aprendizaje automático

El modelo perceptrón

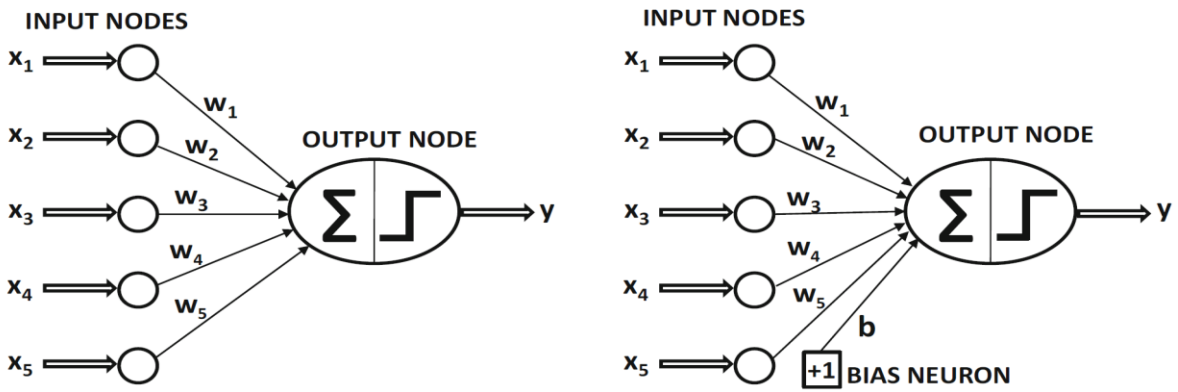
Dr. Omar Mendoza Montoya

04/09/2024

El modelo perceptrón

- La arquitectura más simple de red neuronal es el modelo perceptrón.

- El perceptrón consiste en una sola neurona en su capa oculta, mientras que la función de activación para la salida es la función escalón o umbral.



(a) Perceptron without bias (b) Perceptron with bias

Figure 1.3: The basic architecture of the perceptron

Aggarwal, 2023

El modelo perceptrón

- Este modelo es un clasificador de dos clases.

- Para una observación $X = [x_1\ x_2\ \ldots\ x_n]^T$, devuelve una etiqueta $\hat{y} \in \{-1,1\}$ de acuerdo a la siguiente expresión:

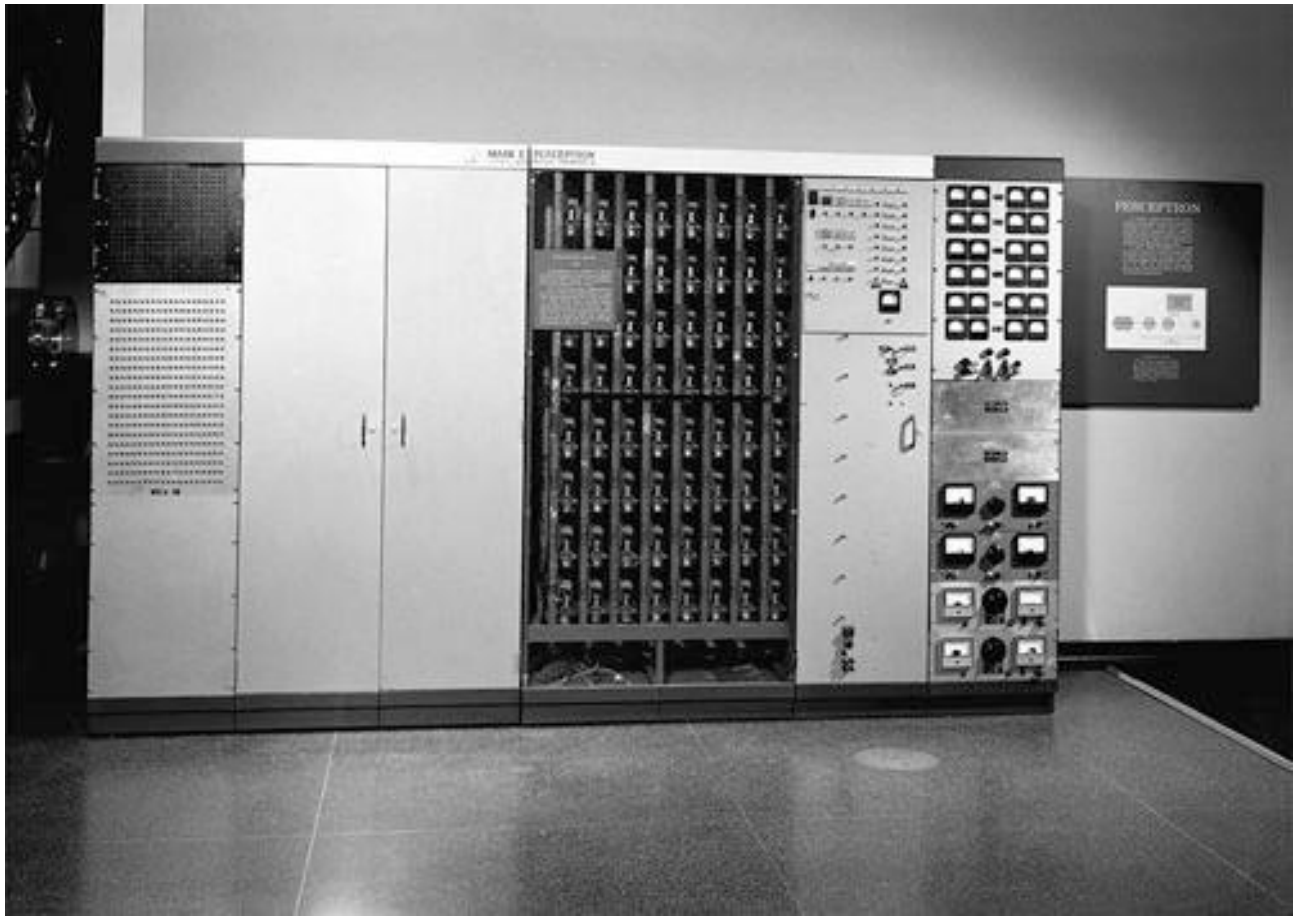
$$\hat{y} = \text{sign}(W^T X) = \text{sign}\left(\sum_{i=1}^{n} w_i x_i\right)$$

donde $W = [w_1\ w_2\ \ldots\ w_n]^T$, y $\text{sign}(\text{x})$ es la función signo, es decir:

$$\text{sign}(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x = 0 \\ -1 & \text{si } x < 0 \end{cases}$$

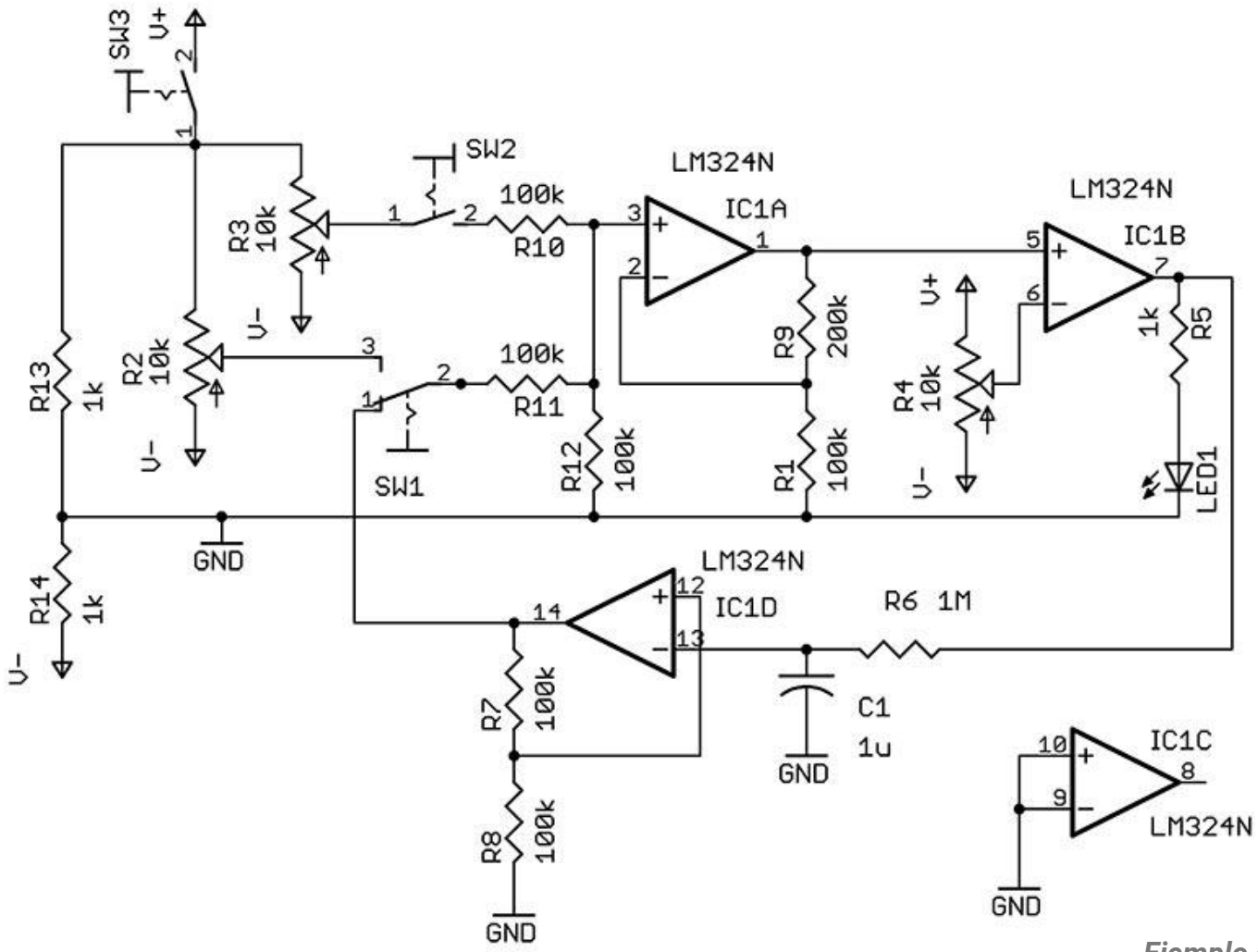
El modelo perceptrón

- En un principio, el perceptrón se implementó con circuitería analógica.



Sistema Mark I Perceptron construido en 1958. Cortesía de Smithsonian Institude

Entrenamiento del perceptrón



Ejemplo de un circuito perceptrón
The perceptron circuit

Entrenamiento del perceptrón

- Para ajustar los pesos, se utilizó una heurística en la que los valores de los pesos se actualizan de acuerdo con la siguiente regla:

$$W \Leftarrow W + \alpha(y - \hat{y})X$$

donde y es la verdadera etiqueta de una observación X.

- A esta expresión se le conoce como la regla de aprendizaje del perceptrón.

- Cuando un dato es clasificado correctamente por el perceptrón, los pesos no se actualizan. Por el contrario, cuando un dato es clasificado incorrectamente, los pesos se actualizan.

Entrenamiento del perceptrón

- Para cada una de las observaciones en el conjunto D, se repite la regla. Incluso se pueden hacer varias pasadas por todo el conjunto de entrenamiento.

- Si los datos son linealmente separables, la regla converge y los pesos dejan de cambiar. Si los datos no son linealmente separables, la regla nunca converge.

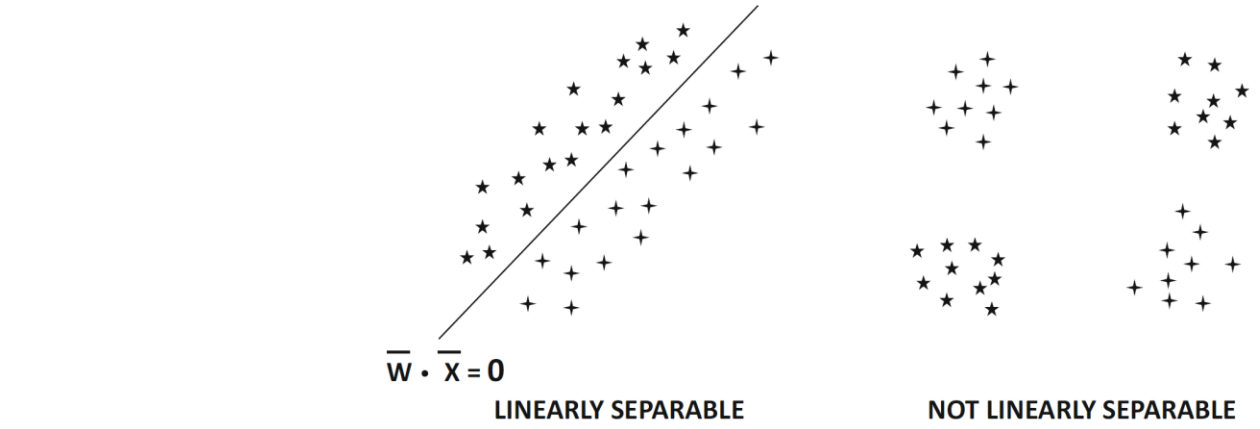


Figure 1.4: Examples of linearly separable and inseparable data in two classes

Aggarwal, 2023

Entrenamiento del modelo perceptrón

Dado un conjunto de datos $D = \{(X_1, y_1), (X_2, y_2), \ldots, (X_n, y_n)\}$, y una constante α pequeña, repetir hasta convergencia:

1. Reordena el conjunto D para obtener el conjunto D^*.
2. Por cada elemento (X_j, y_j) en D^*:

$$W \Leftarrow W + \alpha \left(y_j - \text{sign}\left(W^T X_j\right)\right) X_j$$

Entrenamiento del modelo perceptrón

- Cada actualización del modelo se conoce como lote (Batch), mientras que se le llama época (Epoch) a cada pasada completa con todos los datos de entrenamiento.
- Como suelen seleccionarse los datos que alimentan al perceptrón de manera aleatoria, el algoritmo se conoce como Descenso de Gradiente Estocástico.
- En descenso de gradiente estocástico, el tamaño del lote (Batch size) es 1, ya que por cada dato se hace una actualización de los pesos.

¿Por qué funciona la regla del perceptrón?

Regla de aprendizaje del perceptrón

- La regla del perceptrón funciona como una variación de descenso de gradiente para una función de costo conocida como L_i^{SMOOTH}.

$$L_i^{\text{SMOOTH}} = \max(-y_i W^T X_i, 0)$$

- Esta función penaliza cuando la etiqueta y_i tiene diferente signo que $W'X_i$, es decir, cuando la etiqueta $\hat{y}_i$ es diferente a y_i.

- El gradiente para dicha función de costo está dado por:

$$\nabla L_i^{\text{SMOOTH}} = -(y_i - \hat{y}_i)\bar{X}$$

A diferencia de utilizar todo un conjunto de datos para hacer una sola actualización de los pesos del perceptrón considerando la suma de todos los costos L_i^{SMOOTH}, en la regla del perceptrón se hace una actualización por cada dato del conjunto.

Aunque esto es diferente a como se trabaja en varios modelos clásicos, este enfoque se puede aplicar también a dichos modelos, o incluso hay otras formas de trabajar con el conjunto de datos.

Estrategias para el aprendizaje del perceptrón

- Descenso de gradiente estocástico (stochastic gradiente descent). Se actualiza los pesos del modelo con cada muestra u observación en el conjunto de entrenamiento, las cuales son ordenadas de manera aleatoria.

- Descenso de gradiente de lote (batch gradiente descent). Se calcula la suma de todos los gradientes ∇L_i para solo hacer una actualización del modelo con todos los datos.

$$\nabla L(D) = \sum_{i=1}^{n} \nabla L_i$$

Estrategias para el aprendizaje del perceptrón

- Descenso de gradiente de mini lote (mini-batch gradiente descent). Se calcula la suma de los gradientes ∇L_i de un pequeño grupo de observaciones para solo hacer una actualización del modelo.

$$\nabla L_B = \sum_{i \in B} \nabla L_i$$

donde B es el conjunto de observaciones del lote.

Variantes del modelo perceptrón

Variantes del modelo perceptrón

Tipo de modelo	Función de costo	Regla de aprendizaje
Perceptrón (salida discreta)	$\left(y - \text{sign}(W^T X)\right)^2$	$W \Leftarrow W + \alpha(y - \hat{y})X$
Perceptrón (salida continua)	$\max(0, -yW^T X)$	$W \Leftarrow W + \alpha(y - \hat{y})X$
Regresión lineal	$(y - W^T X)^2$	$W \Leftarrow W + \alpha(y - \hat{y})X$
Regresión logística	$-\ln\left(\left\|\frac{y}{2} - 0.5 + \hat{y}\right\|\right)$	$W \Leftarrow W + \alpha \frac{yX}{1 + e^{yW^T X}}$
Regresión logística (alternativa)	$\ln\left(1 + e^{-yW^T X}\right)$	$W \Leftarrow W + \alpha \frac{yX}{1 + e^{yW^T X}}$
Máquinas de soporte vectorial	$\max(0, -yW^T X + 1)$	$W \Leftarrow W + \alpha yX[I(y\hat{y} < 1)]$

Variantes del modelo perceptrón

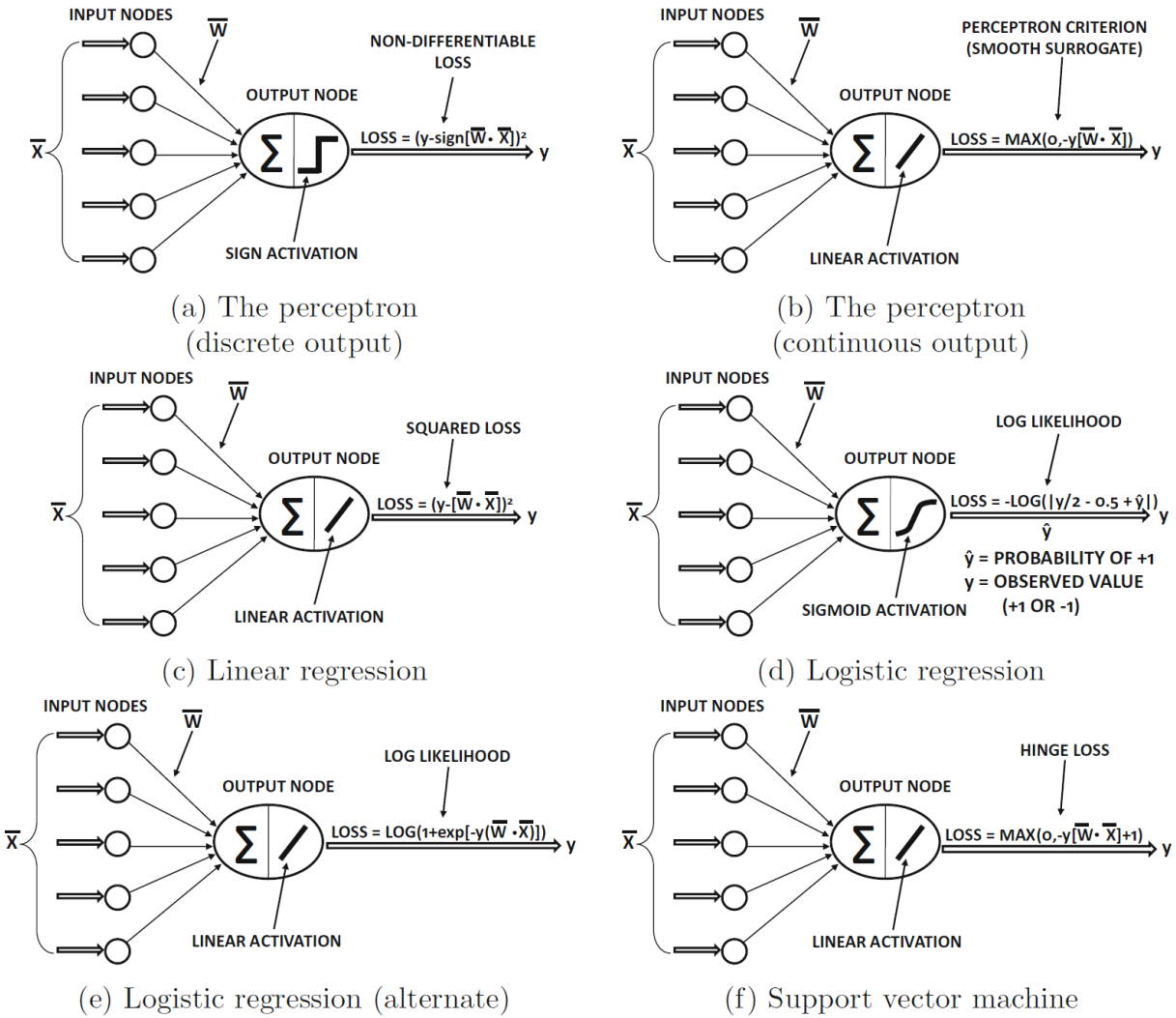


Figure 2.3: Different variants of the perceptron

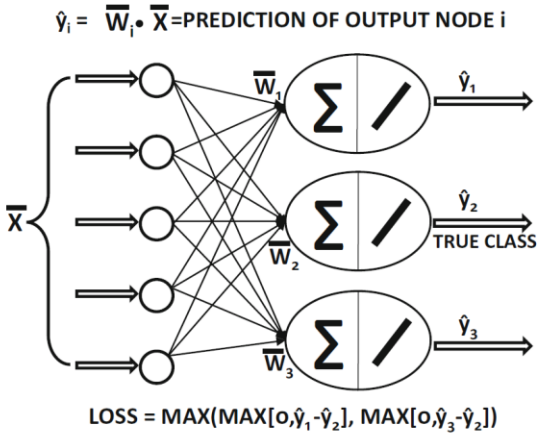
Aggarwal, 2023

Modelos multiclase

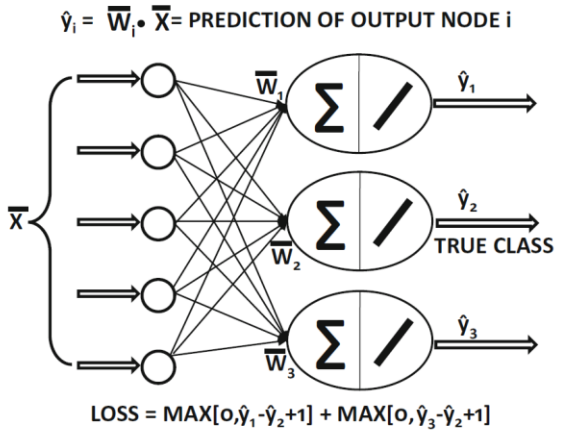
Tipo de modelo	Función de costo	Regla de aprendizaje
Perceptrón multiclase	$\max(\max(0, \hat{y}_1 - \hat{y}_r),\ \max(0, \hat{y}_2 - \hat{y}_r), \ldots, \max(0, \hat{y}_c - \hat{y}_r))$	$W \Leftarrow W + \begin{cases} \alpha X & \text{si } r = \hat{r} \\ -\alpha X & \text{si } r \neq \hat{r} \\ 0 & \text{en otros casos} \end{cases}$
SVM multiclase	$\sum_{i \neq r} \max(0, \hat{y}_i - \hat{y}_r + 1)$	$W \Leftarrow W + \begin{cases} -X\left(\sum_{j \neq r} \delta(j, X)\right) & \text{si } r = \hat{r} \\ X\delta(j, X) & \text{si } r \neq \hat{r} \end{cases}$ $\delta(r, X) =$ función que indica (0 ó 1) cuando el clasificador de la clase r contribuye positivamente a la función de costo
Regresión logística multinomial	$-\ln(\hat{y}_r)$	$W \Leftarrow W + \alpha \begin{cases} X(1 - P(r\|X)) & \text{si } r = \hat{r} \\ -XP(r\|X) & \text{si } r \neq \hat{r} \end{cases}$

$r =$ verdadera clase para la observación X $\quad$ $\hat{r} =$ clase estimada por el modelo $\quad$ $c =$ número de clases

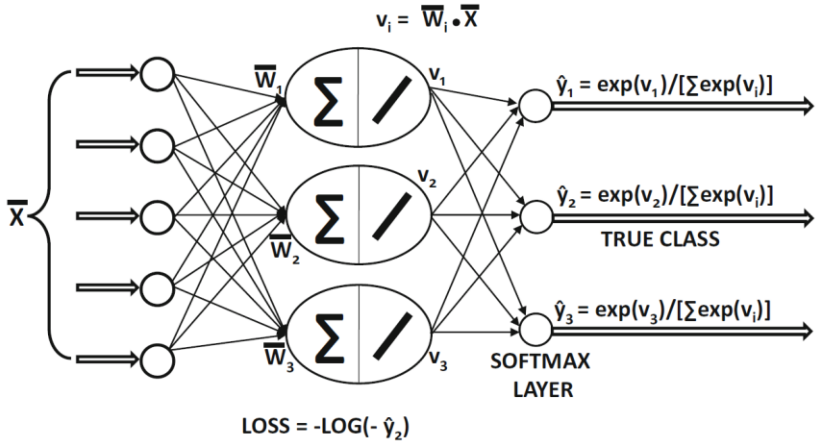
Modelos multiclase



(a) Multiclass perceptron



(b) Multiclass SVM



(c) Multinomial logistic regression

Aggarwal, 2023

Figure 2.5: Multiclass models: In each case, class 2 is assumed to be the ground-truth class.

Bibliografía

- Aggarwal , C. C. (2023). *Neural networks and deep learning* (2da ed.). Springer.
  - Capítulo 1