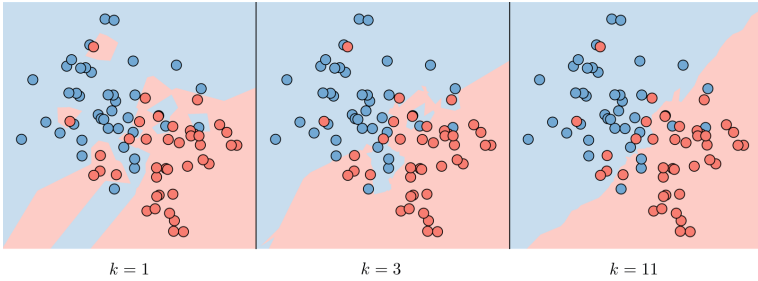


□ **Processus de minimisation de la fonction objectif** – Lors de l’entraînement d’un modèle, on souhaite minimiser la valeur de la fonction objectif évaluée sur l’ensemble d’entraînement :

$$\text{TrainLoss}(w) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} \text{Loss}(x,y,w)$$

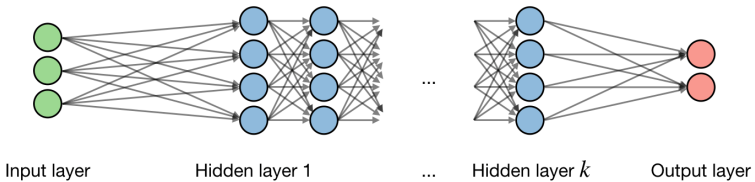
1.3 Prédicteurs non linéaires

□ **k plus proches voisins** – L’algorithme des k plus proches voisins (en anglais *k-nearest neighbors* ou *k-NN*) est une approche non paramétrique où la réponse associée à un exemple est déterminée par la nature de ses k plus proches voisins de l’ensemble d’entraînement. Cette démarche peut être utilisée pour la classification et la régression.



Remarque : plus le paramètre k est grand, plus le biais est élevé. À l’inverse, la variance devient plus élevée lorsque l’on réduit la valeur k .

□ **Réseaux de neurones** – Les réseaux de neurones (en anglais *neural networks*) constituent un type de modèle basés sur des couches (en anglais *layers*). Parmi les types de réseaux populaires, on peut compter les réseaux de neurones convolutionnels et récurrents (abréviés respectivement en CNN et RNN en anglais). Une partie du vocabulaire associé aux réseaux de neurones est détaillée dans la figure ci-dessous :



En notant i la i -ème couche du réseau et j son j -ième neurone, on a :

$$z_j^{[i]} = w_j^{[i]T} x + b_j^{[i]}$$

où l’on note w , b , x , z le coefficient, le biais ainsi que la variable de sortie respectivement.

1.4 Algorithme du gradient stochastique

□ **Descente de gradient** – En notant $\eta \in \mathbb{R}$ le taux d’apprentissage (en anglais *learning rate* ou *step size*), la règle de mise à jour des coefficients pour cet algorithme utilise la fonction objectif