Régression linéaire

La **régression linéaire** est un statistique utilisée pour trouver la relation entre les variables. Dans un modèle de ML contexte, la régression linéaire trouve la relation **fonctionnalités** et **étiquette**.Par exemple, supposons que nous voulions prédire la consommation de carburant d'une voiture en miles par gallon en fonction du poids de la voiture, et nous avons le jeu de données suivant:

Livres en milliers (fonctionnalité)	Miles par gallon (libellé)
3.5	18
3,69	15
3,44	18
3,43	16
4,34	15
4,42	14
2,37	24

Si nous traçions ces points, nous obtiendrions le graphique suivant:

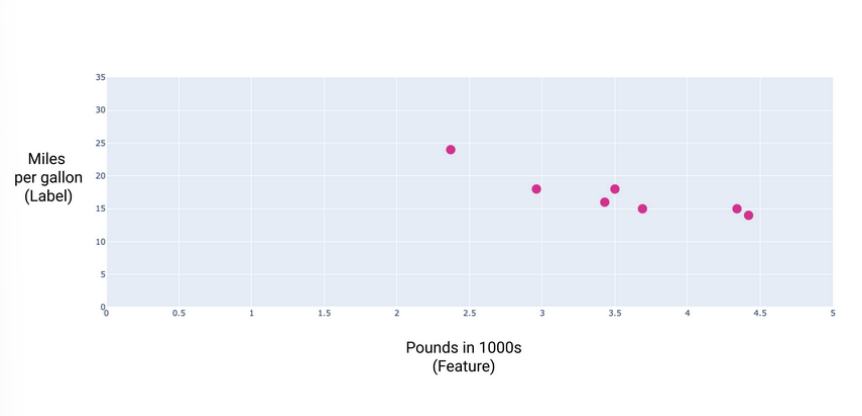


Figure 1 : Poids de la voiture (en livres) par rapport à la classification en miles par gallon. En tant que
la voiture devient plus lourde, sa puissance en miles par gallon diminue généralement.

Nous pourrions créer notre propre modèle en traçant la ligne la mieux adaptée entre les points:

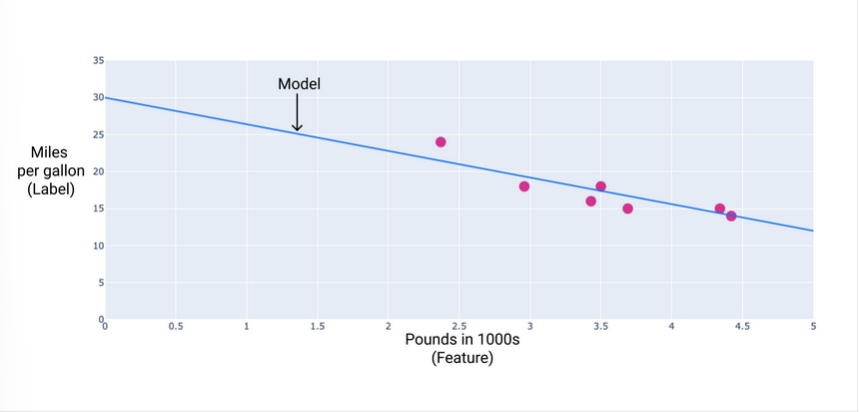


Figure 2 : Une ligne d'ajustement optimale tracée à travers les données de la figure précédente.

Équation de régression linéaire

En termes algébriques, le modèle serait défini comme suit : $y = mx + b$, où

- y est "miles par gallon", c'est-à-dire la valeur que nous voulons prédire.
- m est la pente de la droite.
- x correspond à des livres, soit notre valeur d'entrée.
- b est l'ordonnée à l'origine.

En ML, nous écrivons l'équation d'un modèle de régression linéaire comme suit:

$$y' = b + w_1 x_1$$

où :

- y' est l'étiquette prédite, c'est-à-dire la sortie.
- b correspond au **biais** du modèle. Le biais est le même concept que l'ordonnée à l'origine en algébrique pour une droite. En ML, il est parfois appelé w_0. Biais est un **paramètre** du modèle. est calculé pendant l'entraînement.
- w_1 est la **pondération** de la caractéristique. Dans l'algorithme algébrique, la pondération est le même concept que la pente m pour une droite. La pondération est un **paramètre** du modèle et est calculé pendant l'entraînement.
- x_1 est une **caractéristique** : la saisie.

Pendant l'entraînement, le modèle calcule la pondération et le biais permettant d'obtenir du modèle de ML.

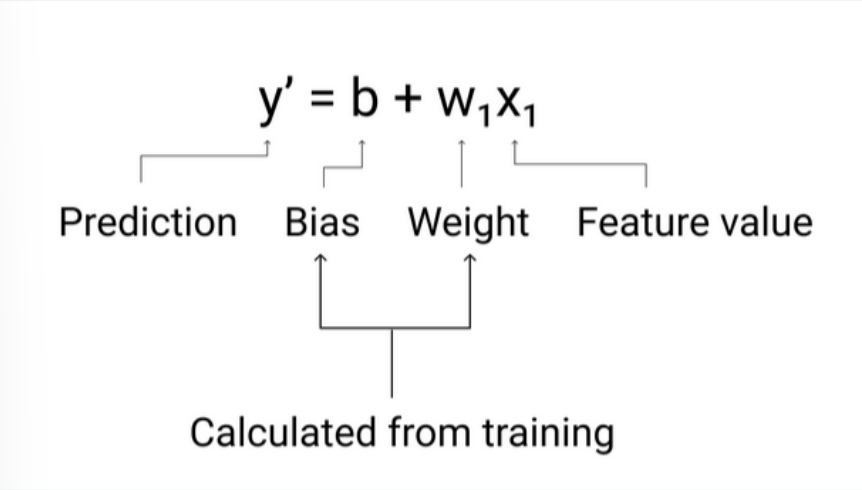


Figure 3. Représentation mathématique d'un modèle linéaire.

Dans notre exemple, nous calculerions le poids et le biais à partir de la ligne que nous avons tracée. La
est de 30 (la ligne croise l'axe des y) et la pondération est de -3,6 (la ligne
la pente de la droite). Le modèle serait alors défini comme suit :

$$y' = 30 + (-3,6)(x_1)$$

et nous pourrions l'utiliser pour faire des prédictions. Par exemple, en utilisant ce modèle, La voiture de 1 800 kilos aurait une consommation de carburant estimée de 25,3km/s.
gallon.

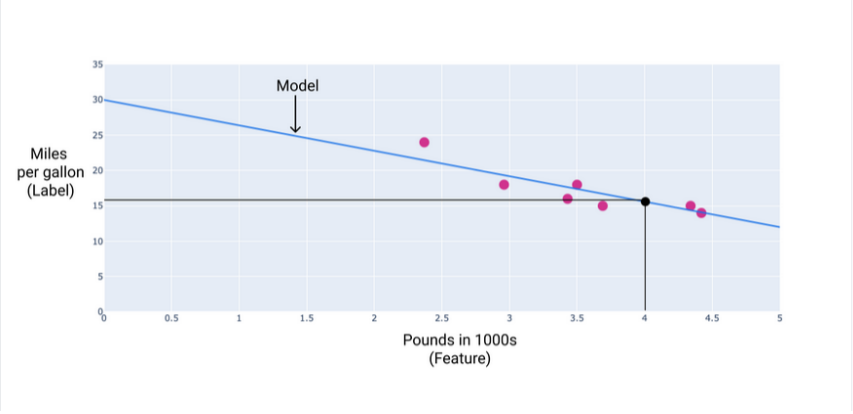


Figure 4. À partir de ce modèle, une voiture de 1 800 kg a une prédiction de carburant de 15,6 miles par gallon.

Modèles avec plusieurs caractéristiques

Bien que l'exemple de cette section n'utilise qu'une seule caractéristique : la lourdeur de la voiture. Un modèle plus sophistiqué peut reposer sur plusieurs caractéristiques, chacune ayant une pondération distincte (w_1, w_2, etc.). Par exemple, un modèle qui repose sur cinq caractéristiques serait écrit comme suit:

$$y' = b + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5$$

Par exemple, un modèle qui prédit la consommation de carburant peut aussi utiliser des caractéristiques
par exemple:

- Cylindrée
- Accélération
- Nombre de cylindres
- Cheval-vapeur anglais

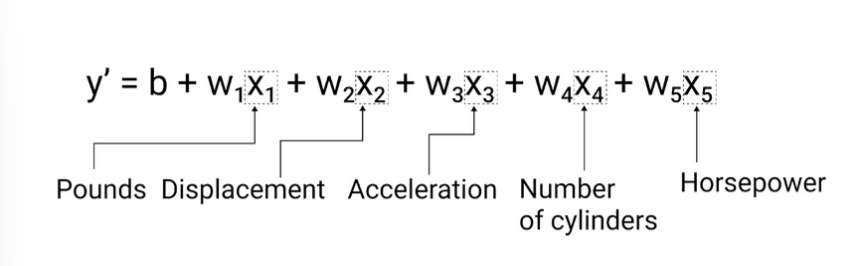


Figure 5. Modèle avec cinq caractéristiques permettant de prédire les miles par gallon d'une voiture
évaluation.
En créant un graphique pour certaines de ces caractéristiques supplémentaires, nous pouvons voir qu'elles ont également relation linéaire à l'étiquette (miles par gallon):

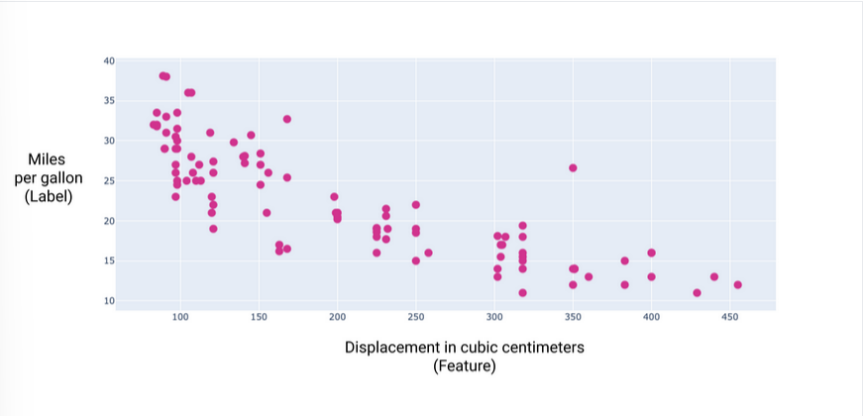


Figure 6. Déplacement d'une voiture en centimètres cubes et en miles par gallon évaluation. À mesure que le moteur d'une voiture devient plus gros, sa capacité en miles par gallon
diminue.

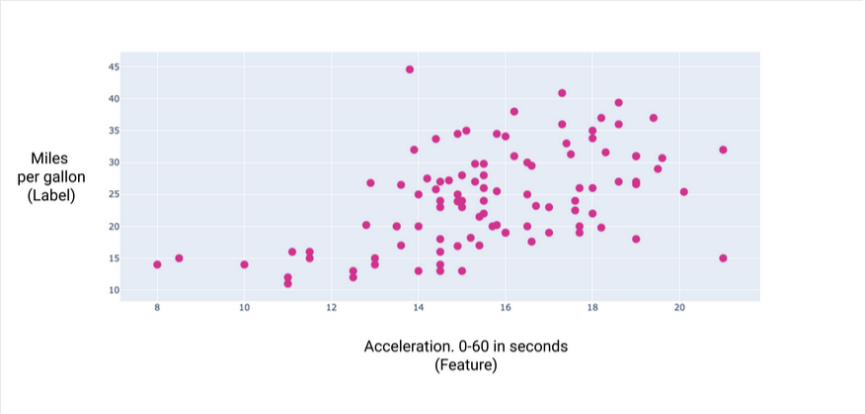


Figure 7 : L'accélération d'une voiture et sa puissance en miles par gallon. Comme une voiture l'accélération prend plus de temps, la cote en miles par gallon augmente généralement.

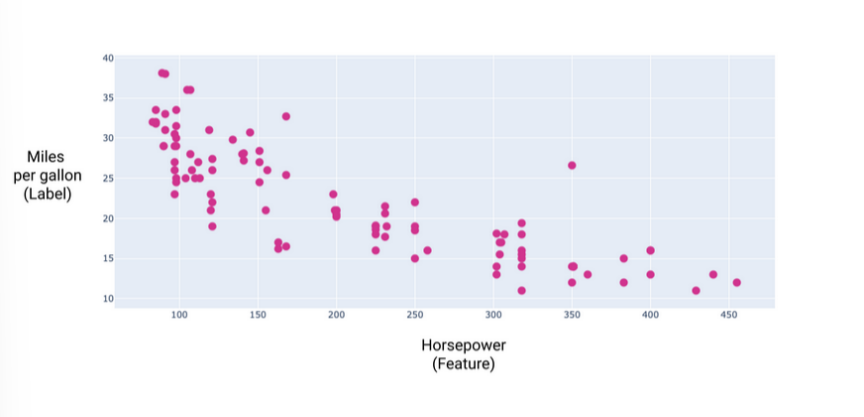


Figure 8. La puissance en chevaux d'une voiture et sa puissance en miles par gallon Comme une voiture la puissance en chevaux augmente, la note en miles par gallon diminue généralement.

Exercice: tester vos connaissances

Quelles parties de l'équation de régression linéaire sont mises à jour pendant l'entraînement ?

- La prédiction ☐
- Les valeurs des caractéristiques ☐
- Les biais et les pondérations ✓

Pendant l'entraînement, le modèle met à jour en fonction de la perte.

Réponse correcte.

▼ key words

weight

Valeur qu'un modèle multiplie par une autre valeur. L'**entraînement** consiste à déterminer les pondérations idéales d'un modèle. L'**inférence** consiste à utiliser ces pondérations apprises pour effectuer des prédictions.

Cliquez sur l'icône pour voir un exemple de pondérations dans un modèle linéaire.

Imaginons un **modèle linéaire** avec deux caractéristiques. Supposons que l'entraînement détermine les poids (et le biais) suivants :

- Le biais, b, a une valeur de 2,2.
- La pondération, w_1 associée à une caractéristique est de 1,5.
- Le poids, w_2, associé à l'autre caractéristique est de 0,4.

Imaginons maintenant un exemple avec les valeurs de fonctionnalités suivantes:

- La valeur d'une caractéristique, x_1, est 6.
- La valeur de l'autre caractéristique, x_2, est 10.

Ce modèle linéaire utilise la formule suivante pour générer une prédiction, y':

$$y' = b + w_1x_1 + w_2x_2$$

La prédiction est donc la suivante:

$$y' = 2.2 + (1.5)(6) + (0.4)(10) = 15.2$$

Si la pondération est égale à 0, la caractéristique correspondante ne contribue pas au modèle. Par exemple, si w_1 est égal à 0, la valeur de x_1 n'a aucune importance.

paramètre

Les **pondérations** et les **biais** qu'un modèle apprend lors de l'**entraînement**. Par exemple, dans un modèle de **régression linéaire**, les paramètres consistent en le biais (*b*) et toutes les pondérations (w_1, w_2, etc.) dans la formule suivante:

$$y' = b + w_1x_1 + w_2x_2 + \ldots w_nx_n$$

À l'inverse, les **hyperparamètres** sont les valeurs que *vous* (ou un service de réglage d'hyperparamètres) fournissez au modèle. Par exemple, le **taux d'apprentissage** est un hyperparamètre.

régression linéaire

Type de modèle de machine learning dans lequel les deux conditions suivantes sont remplies:

- Il s'agit d'un **modèle linéaire**.
- La prédiction est une valeur à virgule flottante. (Il s'agit de la partie **régression** de la *régression linéaire*.)

étiquette

Dans l'**apprentissage automatique supervisé**, partie "réponse" ou "résultat" d'un **exemple**.

Chaque **exemple étiqueté** se compose d'une ou de plusieurs **caractéristiques** et d'un libellé. Par exemple, dans un ensemble de données de détection de spam, l'étiquette sera probablement "spam" ou "non spam". Dans un ensemble de données sur les précipitations, le libellé peut être la quantité de pluie tombée pendant une certaine période.

exemple étiqueté

Exemple contenant une ou plusieurs **caractéristiques** et un **libellé**. Par exemple, le tableau suivant présente trois exemples avec étiquette issus d'un modèle d'évaluation de maison, chacun avec trois caractéristiques et un libellé:

Nombre de chambres	Nombre de salles de bain	Âge de la maison	Prix de la maison (libellé)
3	2	15	345 000 $
2	1	72	179 000 $
4	2	34	392 000 $

Dans le **machine learning supervisé**, les modèles sont entraînés sur des exemples étiquetés et effectuent des prédictions sur des **exemples non étiquetés**.

Comparez un exemple étiqueté à des exemples non étiquetés.

fonctionnalité

Variable d'entrée d'un modèle de machine learning. Un **exemple** se compose d'une ou de plusieurs entités. Par exemple, supposons que vous entraîniez un modèle pour déterminer l'influence des conditions météorologiques sur les résultats des élèves aux tests. Le tableau suivant présente trois exemples, chacun contenant trois caractéristiques et un libellé:

Fonctionnalités			Libellé
Température	Humidité	Pression	Note du test
15	47	998	92
19	34	1020	84
18	92	1012	87

À comparer au **libellé**.

Pour en savoir plus, consultez la section Apprentissage supervisé du cours "Introduction au machine learning".

biais (mathématiques) ou terme de biais

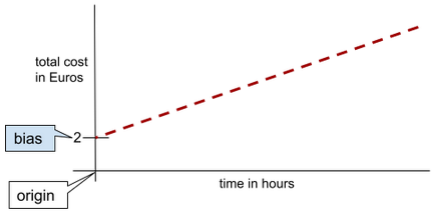
Ordonnée à l'origine ou décalage par rapport à une origine. Le biais est un paramètre des modèles de machine learning, symbolisé par l'un des éléments suivants:

- b
- w_0

Par exemple, b représente le biais dans la formule suivante:

$$y' = b + w_1 x_1 + w_2 x_2 + \ldots w_n x_n$$

Dans une ligne bidimensionnelle simple, le biais correspond simplement à l'ordonnée à l'origine. Par exemple, le biais de la ligne de l'illustration suivante est de 2.



Un biais existe, car tous les modèles ne partent pas de l'origine (0,0). Par exemple, supposons qu'un parc d'attractions coûte 2 euros à l'entrée et 0,5 euro supplémentaire par heure de présence d'un client. Par conséquent, un modèle mappant le coût total présente un biais de 2, car le coût le plus bas est de 2 euros.

Le biais ne doit pas être confondu avec le **biais en matière d'éthique et d'équité** ou le **biais de prédiction**.

Pour en savoir plus, consultez la section Régression linéaire du cours d'initiation au machine learning.

Régression linéaire: perte

La **perte** est une métrique numérique qui décrit l'écart entre les **prédictions** d'un modèle et la réalité. La perte mesure la distance entre les prédictions du modèle et les étiquettes réelles. L'objectif de l'entraînement d'un modèle est de minimiser la perte, en la réduisant à sa valeur la plus basse possible.

Dans l'image suivante, vous pouvez visualiser la perte sous forme de flèches dessinées à partir des points de données vers le modèle. Les flèches indiquent l'écart entre les prédictions du modèle et les valeurs réelles.

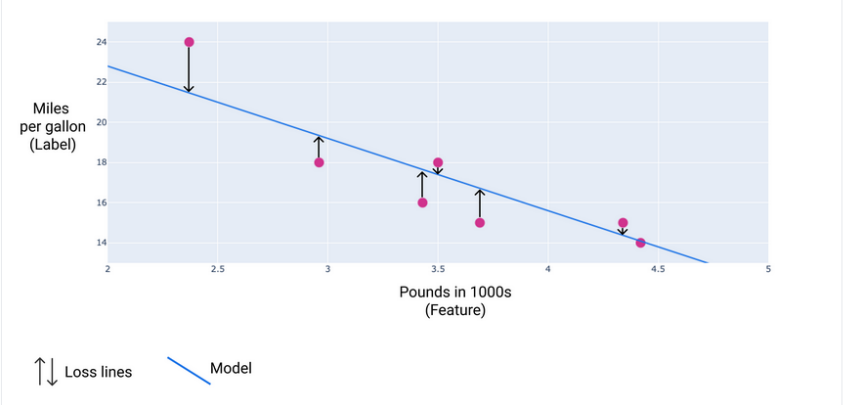


Figure 9. La perte est mesurée entre la valeur réelle et la valeur prédite.

Distance de perte

En statistiques et en apprentissage automatique, la perte mesure la différence entre les valeurs prédites et réelles. La perte se concentre sur la *distance* entre les valeurs, et non sur la direction. Par exemple, si un modèle prédit 2, mais que la valeur réelle est5, peu importe que la perte soit négative -3(2-5=-3).
Au lieu de cela, la
distance entre les valeurs est de 3 . Ainsi, toutes les méthodes de calcul de la perte suppriment le signe.

Voici les deux méthodes les plus courantes pour supprimer le signe:

- Prendre la valeur absolue de la différence entre la valeur réelle et la prédiction.
- Élevez la différence entre la valeur réelle et la prédiction au carré.

Types de pertes

Dans la régression linéaire, il existe quatre principaux types de pertes, qui sont décrits dans le tableau suivant.

Type de perte	Définition	Équation
Perte L_1	Somme des valeurs absolues de la différence entre les valeurs prédites et les valeurs réelles.	$\sum \lvert valeur\ réelle - valeur\ prédite \rvert$
Erreur absolue moyenne (EAM)	Moyenne des pertes L_1 sur un ensemble d'exemples.	$\frac{1}{N} \sum \lvert valeur\ réelle - valeur\ prédite \rvert$
Perte L_2	Somme de la différence au carré entre les valeurs prédites et les valeurs réelles.	$\sum (valeur\ réelle - valeur\ prédite)^2$
Erreur quadratique moyenne (MSE)	Moyenne des pertes L_2 sur un ensemble d'exemples.	$\frac{1}{N} \sum (valeur\ réelle - valeur\ prédite)^2$

La différence fonctionnelle entre la perte L1 et la perte L2
(ou entre l'EAM et l'MSE) est la mise au carré. Lorsque la différence entre la prédiction et le libellé est importante, le calcul au carré rend la perte encore

plus importante. Lorsque la différence est faible (inférieure à 1), la mise au carré réduit encore la perte.

Lorsque vous traitez plusieurs exemples à la fois, nous vous recommandons d'établir la moyenne des pertes pour tous les exemples, que vous utilisiez l'EAM ou la MSE.

Exemple de calcul des pertes

À l'aide de la ligne de meilleure approximation précédente, nous allons calculer la perte L2
pour un seul exemple. À partir de la ligne d'ajustement optimal, nous avons obtenu les valeurs suivantes pour le poids et le biais:

- $Poids : -3,6$
- $Biais : 30$

Si le modèle prédit qu'une voiture de 1 080 kg consomme 11,5 miles par gallon, mais qu'elle consomme en réalité 14,4 miles par gallon, nous calculons la perte L2 comme suit:

Remarque :La formule utilise 2,37, car les graphiques sont mis à l'échelle en milliers de livres.

Valeur	Équation	Résultat
Prédiction	$biais + (poids * caractéristique\ valeur)$ $30 + (-3.6 * 2.37)$	$21,5$
Valeur réelle	$label$	24
perte L_2	$(prédiction - valeurréelle)^2$ $(21.5 - 24)^2$	6.25

Dans cet exemple, la perte L_2 pour ce seul point de données est de 6,25.

Choisir une perte

Le choix de l'EAM ou de MSE dépend de l'ensemble de données et de la manière dont vous souhaitez gérer certaines prédictions. La plupart des valeurs d'éléments d'un ensemble de données se situent généralement dans

une plage distincte. Par exemple, les voitures pèsent généralement entre 2 000 et 5 000 livres et consomment entre 8 et 50 miles par gallon. Une voiture de 8 000 livres ou une voiture qui consomme 100 miles par gallon se situe en dehors de la plage typique et serait considérée comme un **valeur aberrante**.

Un écart peut également faire référence à l'écart entre les prédictions d'un modèle et les valeurs réelles. Par exemple, une voiture de 1 360 kg ou une voiture qui consomme 40 miles par gallon se situent dans les plages typiques. Toutefois, une voiture de 1 360 kg consommant 40 miles par gallon serait un cas atypique par rapport à la prédiction du modèle, car le modèle prédirait qu'une voiture de 1 360 kg consommerait entre 18 et 20 miles par gallon.

Lorsque vous choisissez la meilleure fonction de perte, réfléchissez à la façon dont vous souhaitez que le modèle traite les valeurs aberrantes. Par exemple, la MSE déplace le modèle davantage vers les valeurs aberrantes, contrairement à la MAE. La perte L2 entraîne une pénalité beaucoup plus élevée pour une valeur aberrante que la perte L1.

Par exemple, les images suivantes montrent un modèle entraîné à l'aide de la MAE et un modèle entraîné à l'aide de la MSE. La ligne rouge représente un modèle entièrement entraîné qui servira à effectuer des prédictions. Les anomalies sont plus proches du modèle entraîné avec MSE que du modèle entraîné avec l'EAM.

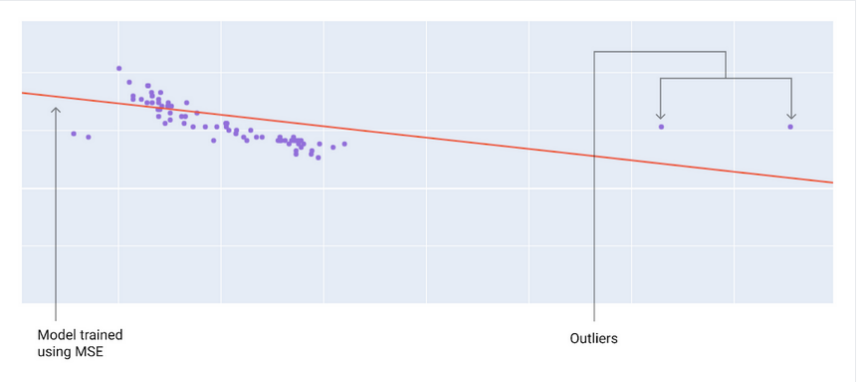


Figure 10 : Un modèle entraîné avec la MSE le rapproche des valeurs aberrantes.

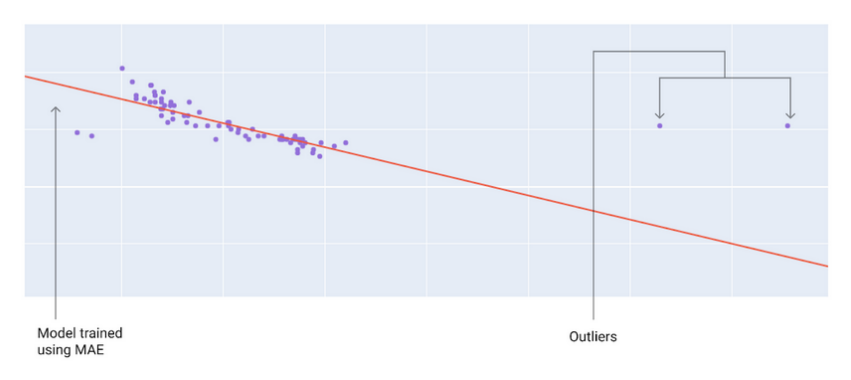
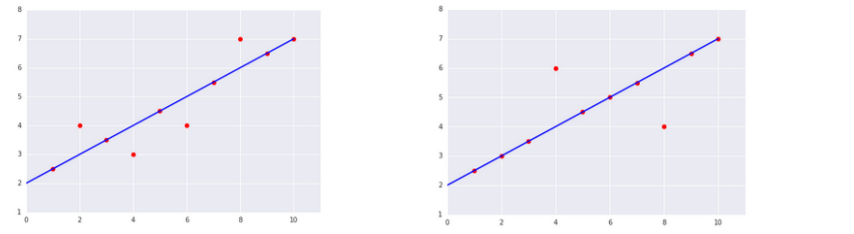


Figure 11 : Un modèle entraîné avec l'EAM est plus éloigné des valeurs aberrantes.

Notez la relation entre le modèle et les données:

- **MSE** (MSE). Le modèle est plus proche des valeurs aberrantes, mais plus éloigné de la plupart des autres points de données.
- **MAE**. Le modèle est plus éloigné des valeurs aberrantes, mais plus proche de la plupart des autres points de données.

Considérons les deux graphiques suivants:



Lequel des deux ensembles de données présentés dans les graphiques précédents a l'erreur quadratique moyenne la plus élevée ?

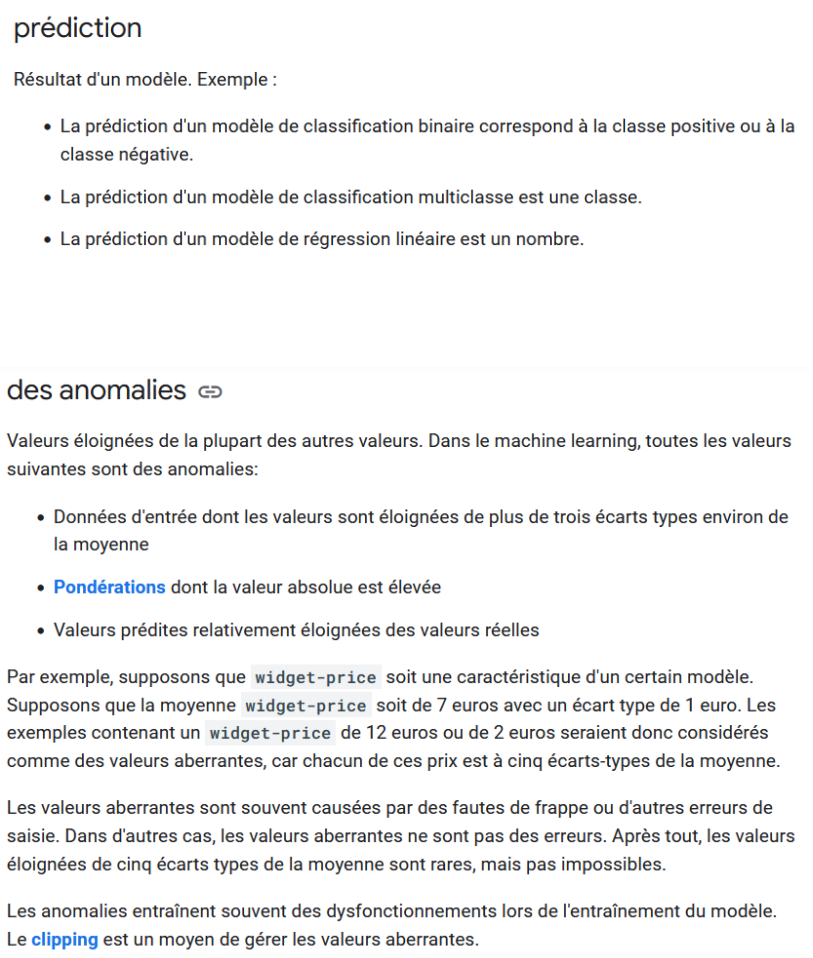
Ensemble de données à droite. ✓

Les huit exemples sur la droite subissent une perte totale de 0. Cependant, bien que seuls deux points se trouvent hors de la ligne, ces deux points sont *deux fois* plus éloignés de la ligne que les points des anomalies sur la figure de gauche. La perte quadratique amplifie ces différences. C'est pourquoi un décalage de deux subit une perte quatre fois plus grande qu'un décalage de un:

$$MSE = \frac{0^2+0^2+0^2+2^2+0^2+0^2+0^2+2^2+0^2+0^2}{10} = 0.8$$

Réponse correcte.

▼ key words



prédiction

Résultat d'un modèle. Exemple :

- La prédiction d'un modèle de classification binaire correspond à la classe positive ou à la classe négative.
- La prédiction d'un modèle de classification multiclasse est une classe.
- La prédiction d'un modèle de régression linéaire est un nombre.

des anomalies

Valeurs éloignées de la plupart des autres valeurs. Dans le machine learning, toutes les valeurs suivantes sont des anomalies:

- Données d'entrée dont les valeurs sont éloignées de plus de trois écarts types environ de la moyenne
- **Pondérations** dont la valeur absolue est élevée
- Valeurs prédites relativement éloignées des valeurs réelles

Par exemple, supposons que `widget-price` soit une caractéristique d'un certain modèle. Supposons que la moyenne `widget-price` soit de 7 euros avec un écart type de 1 euro. Les exemples contenant un `widget-price` de 12 euros ou de 2 euros seraient donc considérés comme des valeurs aberrantes, car chacun de ces prix est à cinq écarts-types de la moyenne.

Les valeurs aberrantes sont souvent causées par des fautes de frappe ou d'autres erreurs de saisie. Dans d'autres cas, les valeurs aberrantes ne sont pas des erreurs. Après tout, les valeurs éloignées de cinq écarts types de la moyenne sont rares, mais pas impossibles.

Les anomalies entraînent souvent des dysfonctionnements lors de l'entraînement du modèle. Le **clipping** est un moyen de gérer les valeurs aberrantes.

Régression linéaire: exercice concernant les paramètres

Le graphique ci-dessous représente 20 exemples à partir d'un ensemble de données sur l'économie de carburant, avec la(poids de la voiture en milliers de livres) tracée sur l'axe des x et la(miles par gallon) tracée sur l'axe des y.

Votre tâche:réglez les curseurs **Pondération** et **Biais** au-dessus du graphique pour
trouver le modèle linéaire qui minimise la perte de MSE sur les données.

Questions à envisager:

- Quelle est la MSE la plus basse que vous puissiez atteindre ?

- Quelles valeurs de pondération et de biais ont provoqué cette perte ?

Nous avons pu obtenir une MSE inférieure à 5 avec une pondération de -1,5 et un biais 21.

Régression linéaire: descente de gradient

La
descente de gradient est une technique mathématique qui trouve de manière itérative les
pondérations et les biais qui produisent le modèle présentant la perte la plus faible. La descente de gradient trouve le meilleur poids et le meilleur biais en répétant le processus suivant pour un certain nombre d'itérations définies par l'utilisateur.

Le modèle commence l'entraînement avec des pondérations et des biais aléatoires proches de zéro, puis répète les étapes suivantes:

1. Calculez la perte à l'aide de la pondération et du biais actuels.
2. Déterminez la direction dans laquelle déplacer les poids et les biais qui réduisent la perte.

3. Déplacez les valeurs de poids et de biais d'une petite quantité dans la direction qui réduit la perte.
4. Revenez à l'étape 1 et répétez le processus jusqu'à ce que le modèle ne puisse plus réduire la perte.

Le schéma ci-dessous décrit les étapes itératives de la descente de gradient pour identifier les pondérations et les biais à l'origine du modèle présentant la perte la plus faible.

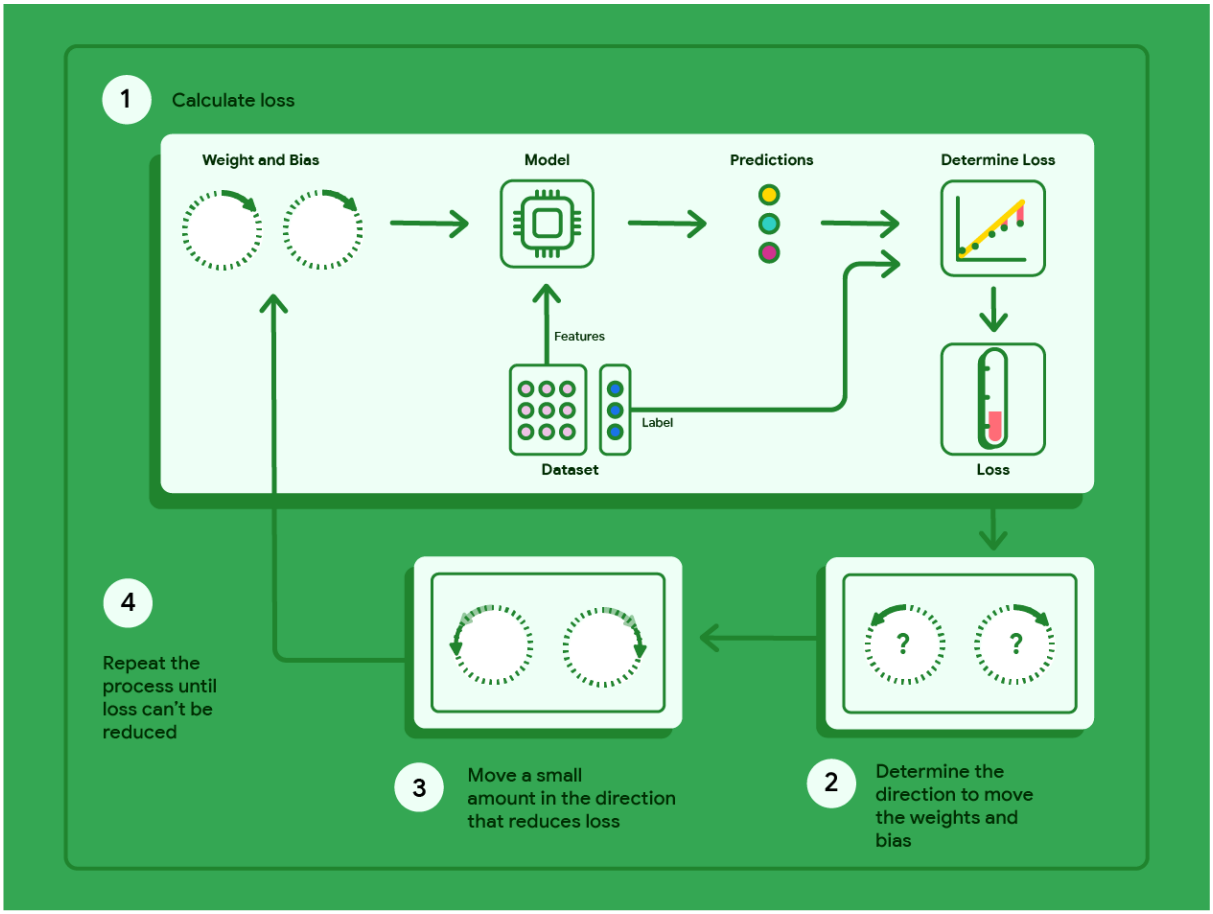


Figure 12. La descente du gradient est un processus itératif qui trouve les pondérations et les biais qui produisent le modèle avec la perte la plus faible.

Concrètement, nous pouvons parcourir les étapes de la descente de gradient à l'aide d'un petit ensemble de données comprenant sept exemples de poids d'une voiture en livres et sa cote en miles par gallon:

Livres par milliers (fonctionnalité)	Miles par gallon (libellé)
3.5	18
3,69	15
3.44	18
3,43	16
4,34	15
4,42	14
2,37	24

1. Le modèle commence l'entraînement en définissant la pondération et le biais sur zéro:

$$Weight: \ 0$$

$$Bias: \ 0$$

$$y = 0 + 0(x_1)$$

2. Calculez la perte MSE avec les paramètres du modèle actuels:

$$Loss = \frac{(18-0)^2 + (15-0)^2 + (18-0)^2 + (16-0)^2 + (15-0)^2 + (14-0)^2 + (24-0)}{7}$$

$$Loss = 303.71$$

3. Calculez la pente de la tangente à la fonction de perte à chaque poids et au biais:

$$Weight\ slope: -119.7$$

$$Bias\ slope: -34.3$$

Pour obtenir la pente des lignes tangentes au poids et au biais, nous prenons la dérivée de la fonction de perte par rapport au poids et au biais, puis nous résolvons les équations.

Nous allons écrire l'équation de prédiction comme suit:
$f_{w,b}(x) = (w * x) + b.$

Nous allons écrire la valeur réelle sous la forme: y
.

Nous allons calculer la MSE à l'aide de la formule suivante:
$\frac{1}{M} \sum_{i=1}^{M} (f_{w,b}(x_{(i)}) - y_{(i)})^2$
où i représente l'exemple d'entraînement i et M le nombre d'exemples.

Dérivée de la pondération

La dérivée de la fonction de perte par rapport à la pondération est écrite comme suit:
$\frac{\partial}{\partial w} \frac{1}{M} \sum_{i=1}^{M} (f_{w,b}(x_{(i)}) - y_{(i)})^2$

et renvoie la valeur suivante:
$\frac{1}{M} \sum_{i=1}^{M} (f_{w,b}(x_{(i)}) - y_{(i)}) * 2x_{(i)}$

Nous commençons par additionner chaque valeur prédite moins la valeur réelle, puis nous la multiplions par deux fois la valeur de l'élément géographique. Nous divisons ensuite la somme par le nombre d'exemples. Le résultat est la pente de la ligne tangente à la valeur de la pondération.

Si nous résolvons cette équation avec une pondération et un biais égaux à zéro, nous obtenons -119,7 pour la pente de la droite.

Dérivée des biais

La dérivée de la fonction de perte par rapport au biais s'écrit comme suit:
$\frac{\partial}{\partial b} \frac{1}{M} \sum_{i=1}^{M} (f_{w,b}(x_{(i)}) - y_{(i)})^2$

et donne la valeur suivante:
$\frac{1}{M} \sum_{i=1}^{M} (f_{w,b}(x_{(i)}) - y_{(i)}) * 2$

Nous commençons par additionner chaque valeur prédite moins la valeur réelle, puis nous la multiplions par deux. Nous divisons ensuite la somme par le nombre d'exemples. Le résultat est la pente de la ligne tangente à la valeur du biais.

Si nous résolvons cette équation avec une pondération et un biais égaux à zéro, nous obtenons -34,3 pour la pente de la droite.

4. Déplacez-vous d'une petite quantité dans la direction de la pente négative pour obtenir le poids et le biais suivants. Pour l'instant, nous allons définir arbitrairement la "petite somme" sur 0,01:

$$New\ weight = old\ weight - (small\ amount * weight\ s$$

$$New\ bias = old\ bias - (small\ amount * bias\ slope)$$

$$New\ weight = 0 - (0.01) * (-119.7)$$

$$New\ bias = 0 - (0.01) * (-34.3)$$

$$New\ weight = 1.2$$

$$New\ bias = 0.34$$

Utilisez la nouvelle pondération et le nouveau biais pour calculer la perte et la répétition. En en terminant le processus pour six itérations, nous obtiendrions les pondérations, biais et pertes suivants:

Itération	Poids	Biais	Perte (MSE)
1	0	0	303,71
2	1.2	0,34	170.67
3	2,75	0.59	67,3
4	3.17	0.72	50,63
5	3,47	0.82	42.1
6	3,68	0,9	37,74

Vous pouvez constater que la perte diminue à chaque mise à jour des pondérations et des biais.Dans cet exemple, nous nous sommes arrêtés après six itérations. En pratique, un modèle est entraîné jusqu'à ce qu'il **converge**.Lorsqu'un modèle converge, les itérations supplémentaires ne réduisent pas davantage la perte, car la descente de gradient a identifié les pondérations et les biais qui minimisent presque la perte.
Si le modèle continue de s'entraîner après la convergence, la perte commence à fluctuer légèrement, car le modèle met constamment à jour les paramètres autour de leurs valeurs les plus basses. Cela peut rendre difficile de vérifier que le modèle a effectivement convergé. Pour confirmer que le modèle a convergé, vous devez poursuivre l'entraînement jusqu'à ce que la perte se stabilise.

Convergence du modèle et courbes de perte

Lorsque vous entraînez un modèle, vous examinez souvent une **courbe de perte** pour déterminer si le modèle a **convergé**.La courbe de perte montre comment la perte change à mesure que le modèle s'entraîne. Voici à quoi ressemble une courbe de perte typique. La perte est représentée sur l'axe des y et les itérations sur l'axe
des x:

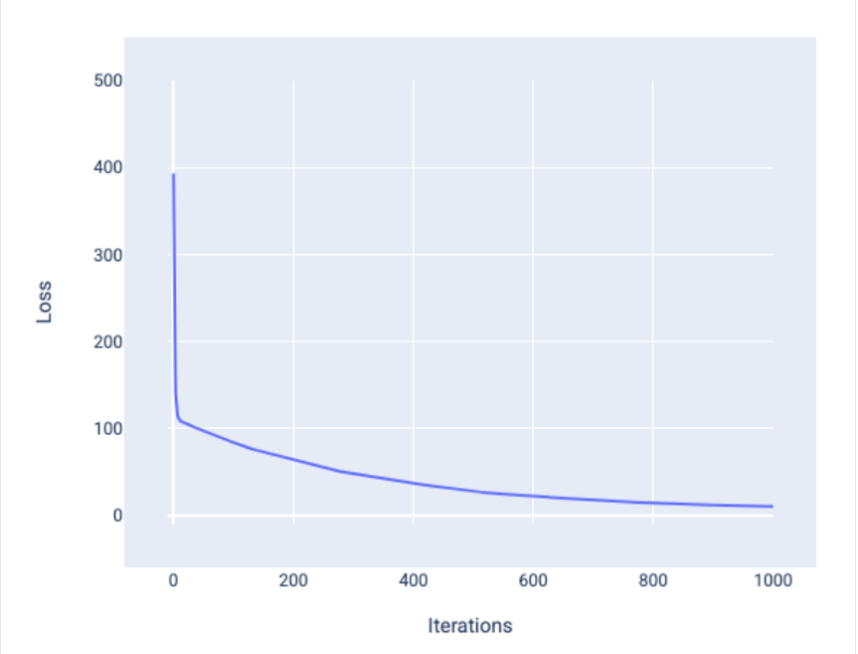


Figure 13. Courbe de perte montrant la convergence du modèle autour de la 1 000e itération.

Vous pouvez constater que la perte diminue considérablement au cours des premières itérations, puis diminue progressivement avant de s'aplatir vers la 1 000e itération. Après 1 000 itérations, nous pouvons être presque certains que le modèle a convergé.

Dans les figures suivantes, le modèle est dessiné à trois points du processus d'entraînement: le début, le milieu et la fin. La visualisation de l'état du modèle lors des instantanés pendant le processus d'entraînement renforce le lien entre la mise à jour des poids et des biais, la réduction des pertes et la convergence du modèle.

Dans les figures, nous utilisons les pondérations et le biais dérivés à une itération donnée pour représenter le modèle. Dans le graphique contenant les points de données et l'instantané du modèle, les lignes de perte bleues entre le modèle et les points de données indiquent la quantité de perte. Plus les droites sont longues, plus il y a de perte.

Dans la figure suivante, nous pouvons voir qu'autour de la deuxième itération, le modèle ne serait pas en mesure de faire de bonnes prédictions en raison de la forte perte.

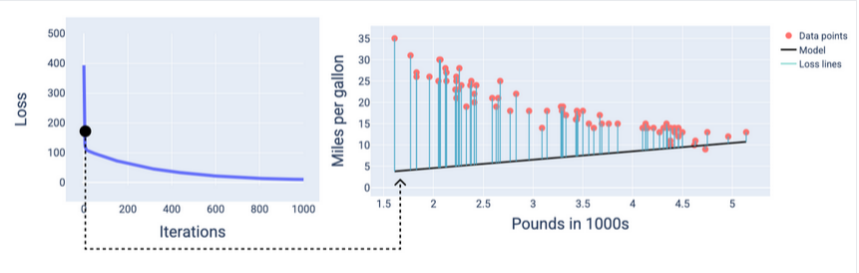


Figure 14. Courbe de perte et instantané du modèle au début du processus d'entraînement.
Vers la 400e itération, nous pouvons voir que la descente du gradient a trouvé les poids et les biais qui produisent un meilleur modèle.

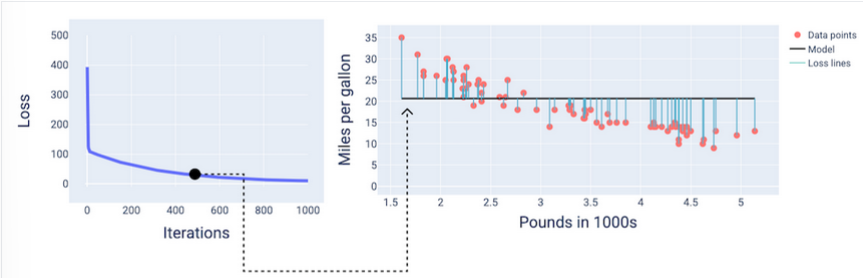


Figure 15. Courbe de perte et instantané du modèle vers le milieu de l'entraînement.
Vers la 1 000e itération, nous pouvons voir que le modèle a convergé, produisant un modèle avec la perte la plus faible possible.

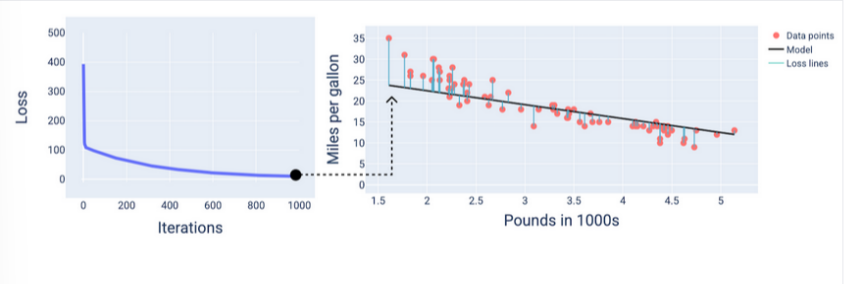


Figure 16 : Courbe de perte et instantané du modèle vers la fin du processus d'entraînement.

Exercice : Vérifiez votre compréhension

Quel est le rôle de la descente du gradient dans la régression linéaire ?

- ☐ La descente du gradient permet de déterminer le type de perte à utiliser lors de l'entraînement d'un modèle, par exemple L_1 ou L_2.
- ☐ La descente du gradient élimine les valeurs aberrantes de l'ensemble de données pour aider le modèle à effectuer de meilleures prédictions.
- ✓ La descente de gradient est un processus itératif qui trouve les meilleures pondérations et biais qui minimisent la perte.

Réponse correcte.

Convergence et fonctions convexes

Les fonctions de perte des modèles linéaires génèrent toujours une surface **convexe**. Grâce à cette propriété, lorsqu'un modèle de régression linéaire converge, nous savons que le modèle a identifié les pondérations et le biais qui génèrent la perte la plus faible.

Si nous créons un graphique du graphe de fonction de perte d'un modèle comportant une seule caractéristique, nous pouvons voir sa forme convexe. Vous trouverez ci-dessous la surface de perte de l'ensemble de données sur les miles par gallon utilisé dans les exemples précédents. Le poids est sur l'axe X, le biais sur l'axe Y et la perte sur l'axe Z:

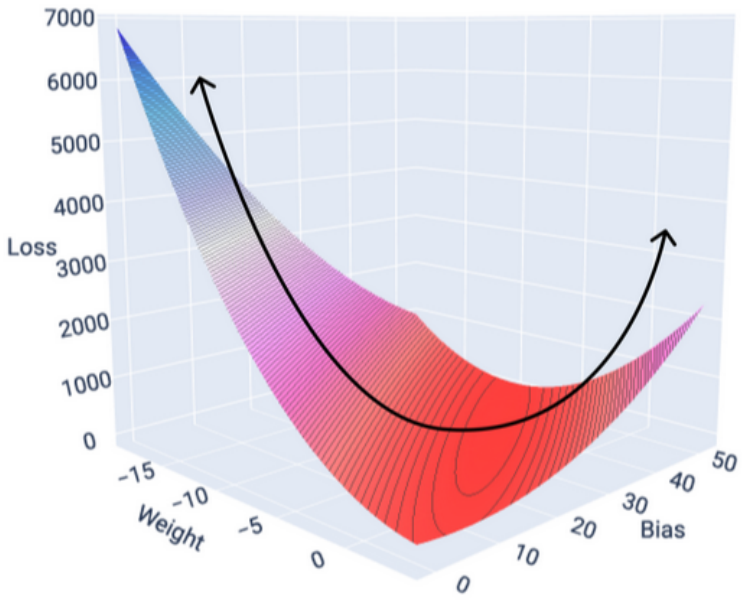


Figure 17. Surface de perte qui montre sa forme convexe.

Dans cet exemple, une pondération de -5,44 et un biais de 35,94 produisent la perte la plus faible à 5,54:

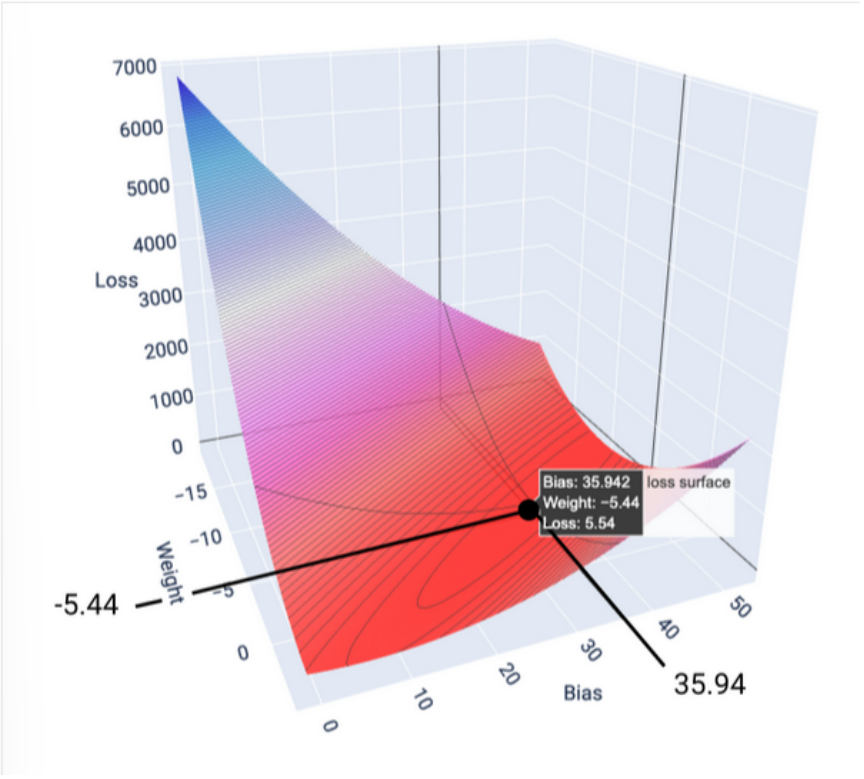


Figure 18 : Surface de perte montrant les valeurs de poids et de biais qui produisent la perte la plus faible.

Un modèle linéaire converge lorsqu'il a trouvé la perte minimale. Par conséquent, les itérations supplémentaires ne font que déplacer les valeurs de poids et de biais de très petites quantités autour du minimum. Si nous créions un graphique des pondérations et des points de biais pendant la descente de gradient, les points ressemblaient à une balle qui descendait une colline et s'arrêtaient au point où il n'y a plus de pente descendante.

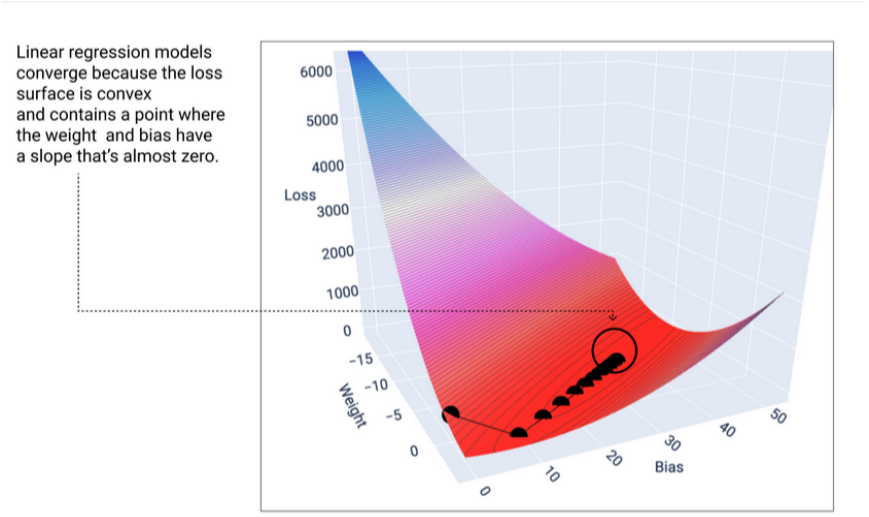


Figure 19 Graphique de perte affichant les points de descente de gradient s'arrêtant au point le plus bas du graphique

Notez que les points de perte noirs créent la forme exacte de la courbe de fonction de perte: une forte baisse avant de baisser progressivement jusqu'à ce qu'ils atteignent le point le plus bas du graphe de fonction de perte.

Il est important de noter que le modèle ne trouve presque jamais le minimum exact pour chaque pondération et biais, mais une valeur très proche. Il est également important de noter que le minimum des poids et du biais ne correspond pas à une perte nulle, mais uniquement à une valeur qui produit la perte la plus faible pour ce paramètre.

En utilisant les valeurs de poids et de biais qui produisent la perte la plus faible (dans ce cas, un poids de -5,44 et un biais de 35,94), nous pouvons représenter le modèle graphiquement pour voir dans quelle mesure il s'adapte aux données:

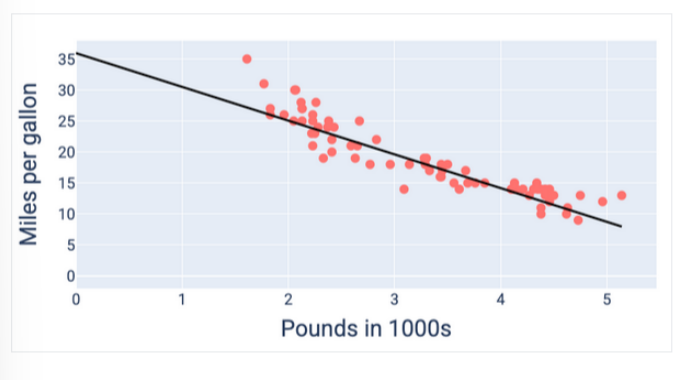


Figure 20. Modèle représenté graphiquement à l'aide des valeurs de pondération et de biais qui génèrent la perte la plus faible.

Régression linéaire: hyperparamètres

Les **hyperparamètres** sont des variables qui contrôlent différents aspects de l'entraînement. Voici trois hyperparamètres courants:

- **Taux d'apprentissage**
- **Taille de lot**
- **Époques**

À l'inverse, les
paramètres sont les variables, comme les pondérations et les biais, qui font partie du modèle lui-même. En d'autres termes, les hyperparamètres sont des valeurs que vous contrôlez, tandis que les paramètres sont des valeurs que le modèle calcule pendant l'entraînement.

Taux d'apprentissage

Le
taux d'apprentissage est un nombre à virgule flottante que vous définissez, qui influence la vitesse
de convergence du modèle. Si le taux d'apprentissage est trop faible, la convergence du modèle peut prendre beaucoup de temps. Toutefois, si le taux

d'apprentissage est trop élevé, le modèle ne converge jamais, mais oscille autour des pondérations et des biais qui minimisent la perte. L'objectif est de choisir un taux d'apprentissage qui n'est ni trop élevé ni trop faible pour que le modèle converge rapidement.

Le taux d'apprentissage détermine l'ampleur des modifications à apporter aux poids et aux biais à chaque étape du processus de descente du gradient. Le modèle multiplie le gradient par le taux d'apprentissage pour déterminer les paramètres du modèle (valeurs de poids et de biais) pour l'itération suivante. Lors de la troisième étape de la
descente de gradient, la "petite quantité" à avancer dans la direction de la pente négative fait référence au taux d'apprentissage.

La différence entre les anciens paramètres du modèle et les nouveaux paramètres du modèle est proportionnelle à la pente de la fonction de perte. Par exemple, si la pente est importante, le modèle effectue un grand pas. Si elle est petite, il faut faire un petit pas. Par exemple, si la magnitude du gradient est de 2,5 et que le taux d'apprentissage est de 0,01, le modèle modifie le paramètre de 0,025.

Le taux d'apprentissage idéal aide le modèle à converger en un nombre raisonnable d'itérations. Dans la figure 21, la courbe de fonction de perte montre que le modèle s'améliore de manière significative au cours des 20 premières itérations avant le début de la convergence:

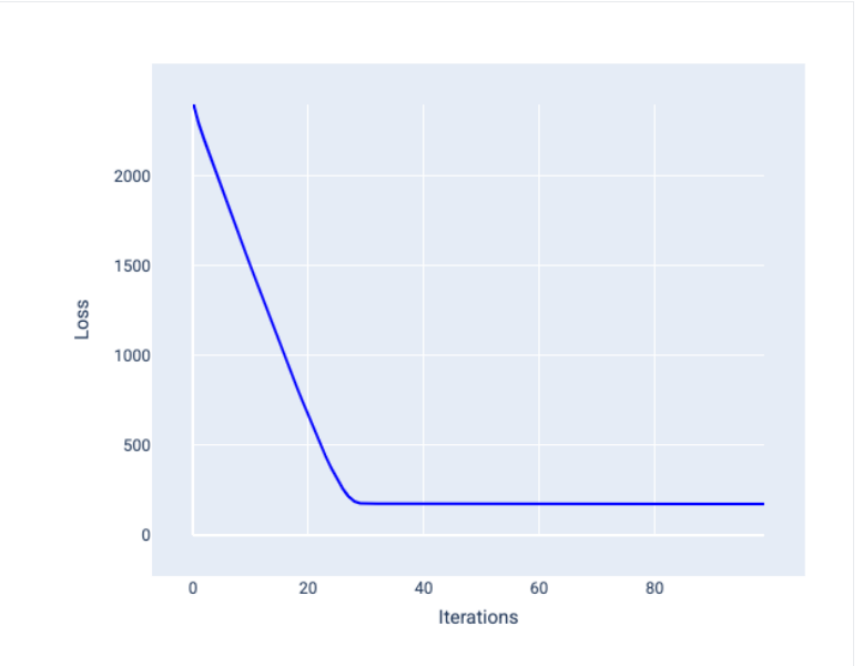


Figure 21. Graphique de perte montrant un modèle entraîné avec un taux d'apprentissage qui converge rapidement.

En revanche, un taux d'apprentissage trop faible peut nécessiter trop d'itérations pour converger. Dans la figure 22, la courbe de fonction de perte montre que le modèle n'apporte que des améliorations mineures après chaque itération:

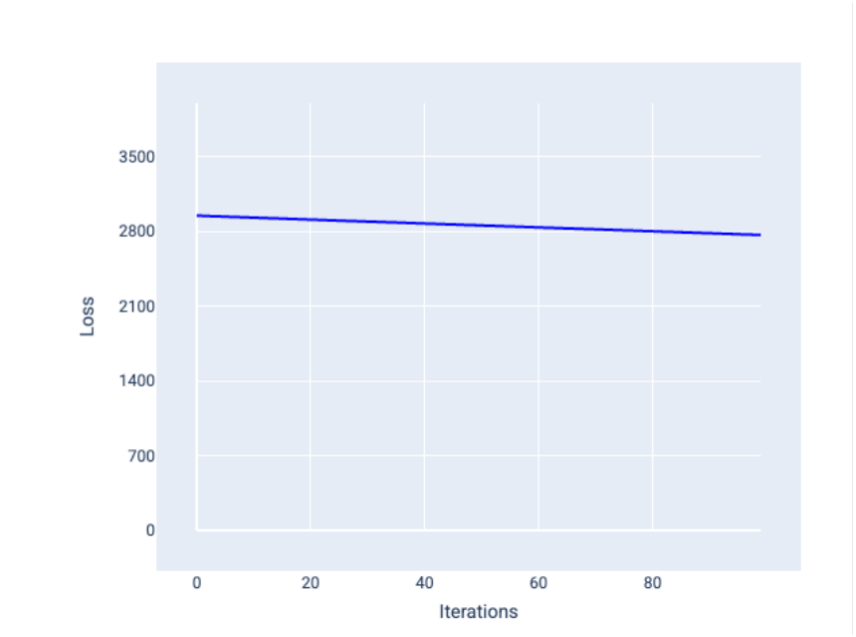


Figure 22 Graphique de perte montrant un modèle entraîné avec un faible taux d'apprentissage.

Un taux d'apprentissage trop élevé ne converge jamais, car chaque itération entraîne soit un rebond de la perte, soit une augmentation continue. Dans la figure 23, la courbe de perte montre que le modèle diminue, puis augmente la perte après chaque itération. Dans la figure 24, la perte augmente lors des itérations ultérieures:

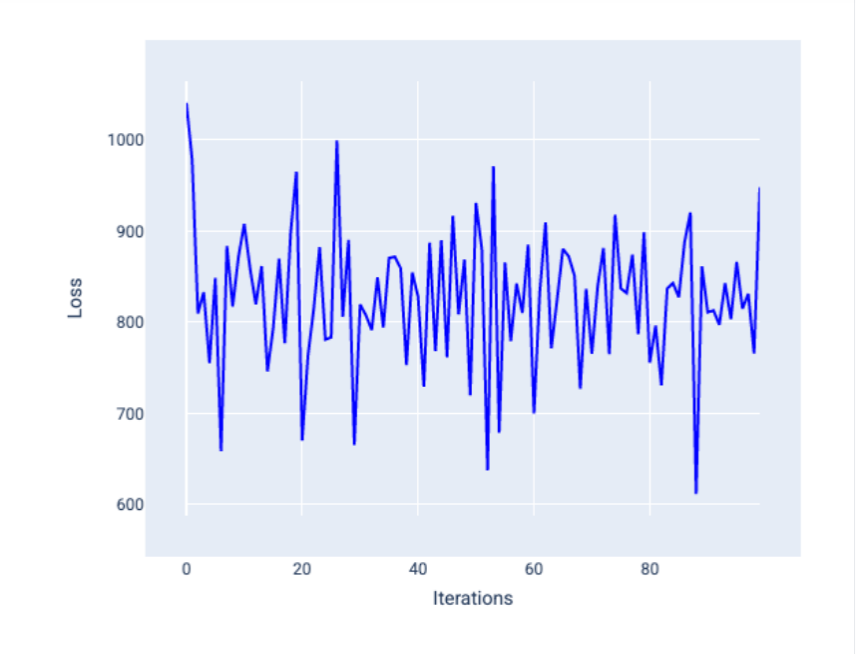
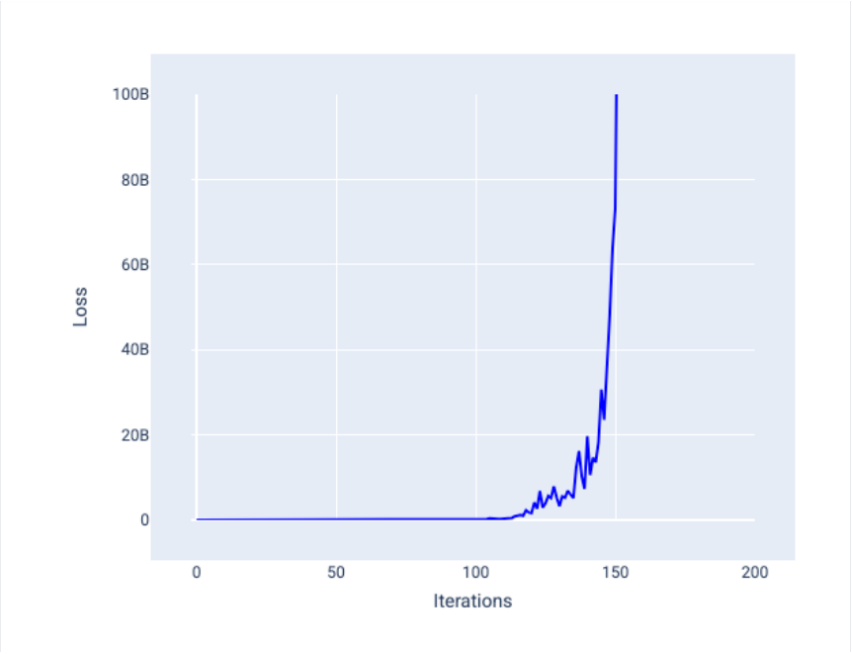


Figure 23. Graphique de perte montrant un modèle entraîné avec un taux d'apprentissage trop élevé, où la courbe de perte fluctue de manière extrême, à la hausse et à la baisse à mesure que les itérations augmentent.



Exercice : Vérifiez votre compréhension

Quelle est la vitesse d'apprentissage idéale ?

- ☐ 0,01
- ☐ 1.0
- ✓ Le taux d'apprentissage idéal dépend du problème.

Chaque modèle et ensemble de données a sa propre vitesse d'apprentissage idéale.

Réponse correcte.

Taille de lot

La **taille de lot** est un hyperparamètre qui fait référence au nombre d'exemples que le modèle traite avant de mettre à jour ses poids et ses biais. Vous pourriez penser que le modèle doit calculer la perte pour *chaque* exemple de l'ensemble de données avant de mettre à jour les poids et le biais. Toutefois, lorsqu'un ensemble de données contient des centaines de milliers, voire des millions d'exemples, il n'est pas pratique d'utiliser l'ensemble complet.

Deux techniques courantes pour obtenir le bon gradient en *moyenne* sans avoir à examiner chaque exemple de l'ensemble de données avant de mettre à jour

les poids et le biais sont la **descente stochastique du gradient** et la **descente stochastique du gradient en mini-lot**:

- **Descente de gradient stochastique (SGD)**: la descente de gradient stochastique n'utilise qu'un seul exemple (une taille de lot d'un) par itération. Avec suffisamment d'itérations, SGD fonctionne, mais est très bruyant. Le terme "bruit" désigne les variations lors de l'entraînement qui entraînent une augmentation plutôt qu'une diminution de la perte lors d'une itération. Le terme"stochastique" indique que l'exemple constituant chaque lot est choisi au hasard.

  Notez dans l'image suivante que la perte fluctue légèrement à mesure que le modèle met à jour ses poids et ses biais à l'aide de la descente de gradient stochastique, ce qui peut entraîner du bruit dans le graphique de la perte:

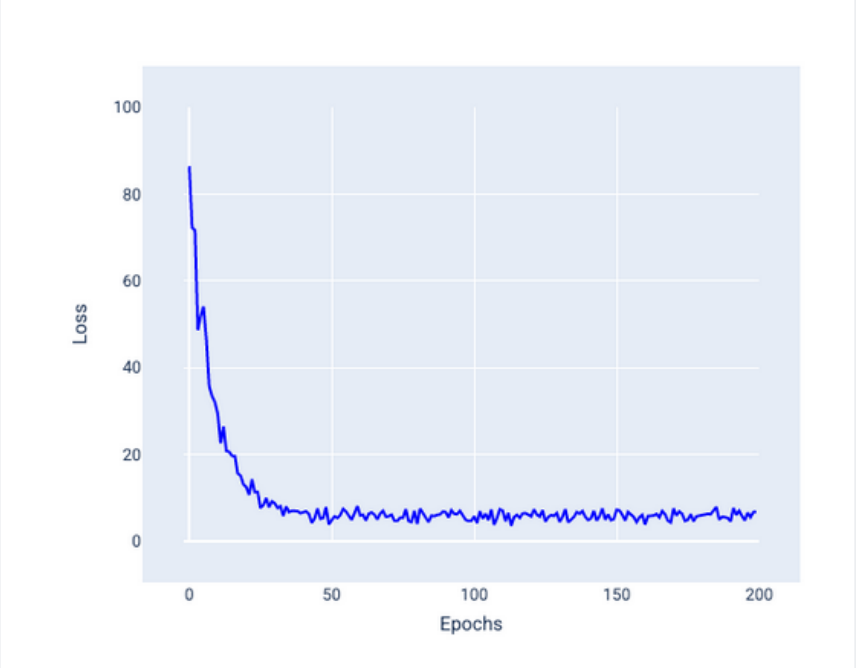


Figure 25 : Modèle entraîné avec une descente de gradient stochastique (SGD) montrant du bruit dans la courbe de perte.

- Notez que l'utilisation de la descente stochastique du gradient peut produire du bruit sur l'ensemble de la courbe de perte, et pas seulement à proximité de la convergence.
- **Descente de gradient stochastique par mini-lots (SGD par mini-lots)**: la descente de gradient stochastique par mini-lots constitue un compromis entre le traitement par lot complet et la SGD. Pour N nombre de points de

données, la taille de lot peut être n'importe quel nombre supérieur à 1 et inférieur à N.
 Le modèle choisit aléatoirement les exemples inclus dans chaque lot, calcule la moyenne de leurs gradients, puis met à jour les pondérations et le biais une fois par itération.

Le nombre d'exemples pour chaque lot dépend de l'ensemble de données et des ressources de calcul disponibles. En général, les petites tailles de lot se comportent comme la descente du gradient stochastique, tandis que les tailles de lot plus importantes se comportent comme la descente du gradient sur l'ensemble du lot

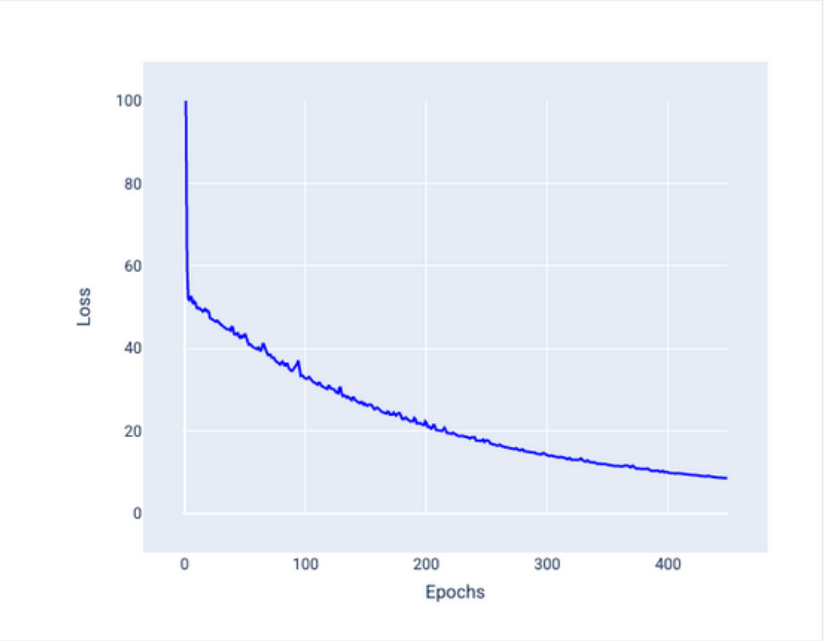


Figure 26 Modèle entraîné avec SGD par mini-lots.

Lorsque vous entraînez un modèle, vous pouvez penser que le bruit est une caractéristique indésirable qui doit être éliminée. Toutefois, un certain niveau de bruit peut être bénéfique. Dans les modules suivants, vous découvrirez comment le bruit peut aider un modèle à mieux **généraliser** et à trouver les poids et les biais optimaux dans un **réseau de neurones**.

Époques

Pendant l'entraînement, une **époque** signifie que le modèle a traité chaque exemple de l'ensemble d'entraînement *une seule fois*. Par exemple, étant donné un ensemble d'entraînement de 1 000 exemples et une taille de mini-lot de 100 exemples, le modèle nécessitera 10 **iterations** pour terminer une époque.L'entraînement nécessite généralement de nombreuses époques. Autrement dit, le système doit traiter plusieurs fois chaque exemple de l'ensemble d'entraînement.

Le nombre d'époques est un hyperparamètre que vous définissez avant le début de l'entraînement du modèle. Dans de nombreux cas, vous devrez tester le nombre d'époques nécessaires pour que le modèle converge. En général, un plus grand nombre d'époques produit un meilleur modèle, mais l'entraînement prend également plus de temps.

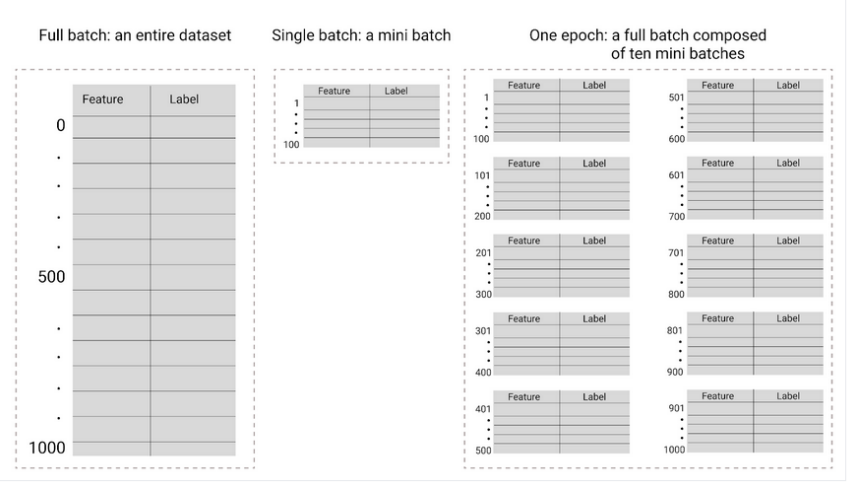


Figure 27 Lot complet par rapport à un mini-lot.

Le tableau suivant décrit la relation entre la taille de lot et les époques et le nombre de fois qu'un modèle met à jour ses paramètres.

Type de lot	Quand les mises à jour des pondérations et des biais se produisent
Lot complet	Une fois que le modèle a examiné tous les exemples de l'ensemble de données, Par exemple, si un ensemble de données contient 1 000 exemples et que le modèle s'entraîne pendant 20 époques, il met à jour les poids et le biais 20 fois, une fois par époque.
Descente de gradient stochastique	Une fois que le modèle a examiné un seul exemple de l'ensemble de données. Par exemple, si un ensemble de données contient 1 000 exemples et s'entraîne pendant 20 époques, le modèle met à jour les pondérations et les biais 20 000 fois.
Descente de gradient stochastique par mini-lot	Une fois que le modèle a examiné les exemples de chaque lot. Par exemple, si un ensemble de données contient 1 000 exemples, que la taille de lot est de 100 et que le modèle s'entraîne pendant 20 époques, il met à jour les poids et le biais 200 fois.

Exercice : Vérifiez votre compréhension

1. Quelle est la meilleure taille de lot à utiliser avec la descente de gradient stochastique par mini-lots ?

- [x] Cela dépend

  La taille de lot idéale dépend de l'ensemble de données et des ressources de calcul disponibles.

  Réponse correcte.

- [ ] 100 exemples par lot
- [ ] 10 exemples par lot

2. Laquelle de ces affirmations est correcte ?

- [x] Doubler le taux d'apprentissage peut ralentir l'entraînement.

  Cette affirmation est vraie. Le doublement du taux d'apprentissage peut entraîner un taux d'apprentissage trop élevé et, par conséquent, provoquer un "rebondissement" des pondérations, augmentant ainsi le temps nécessaire à la convergence. Comme toujours, les meilleurs hyperparamètres dépendent de votre ensemble de données et des ressources de calcul disponibles.

  Réponse correcte.

- [ ] Les lots plus importants ne sont pas adaptés aux données comportant de nombreux écarts.

Régression linéaire: exercice de descente de gradient

Dans cet exercice, vous allez revenir sur le graphique des données d'efficacité énergétique de l'exercice sur les paramètres. Mais cette fois, vous allez utiliser la descente du gradient pour apprendre les valeurs optimales de poids et de biais pour un modèle linéaire qui minimise les pertes.

Réalisez les trois tâches ci-dessous le graphique.