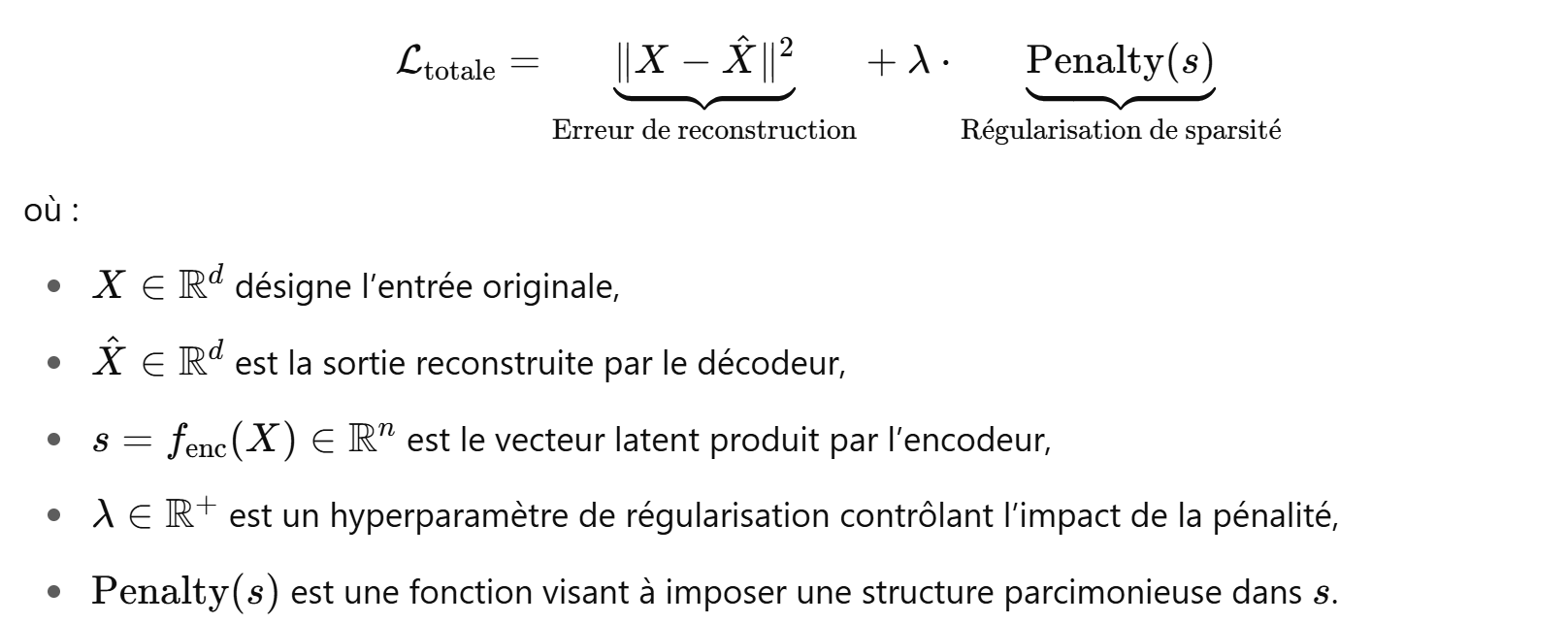
### Sparsité dans les autoencodeurs : formulation et mécanismes

Dans le cadre de la compression de données par autoencodeur, la sparsité du vecteur latent constitue une propriété essentielle permettant de contraindre le modèle à produire des représentations compactes et informatives. L’objectif est de forcer la majorité des activations dans l’espace latent à être nulles ou proches de zéro, tout en préservant les informations clés nécessaires à la reconstruction des données d’entrée. Cette approche est particulièrement pertinente dans des contextes contraints en mémoire et en calcul, tels que les systèmes embarqués.

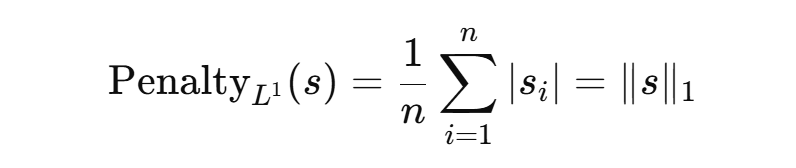
#### Formulation générale de la fonction de perte

La fonction de perte d’un autoencodeur intégrant une régularisation de sparsité s’écrit généralement comme suit :

Deux approches principales sont utilisées pour définir cette fonction de pénalité : la régularisation et la divergence de Kullback-Leibler (KL-divergence).

#### 1. Régularisation (Lasso)

La norme constitue une méthode simple et efficace pour induire la sparsité. Elle est définie comme suit :



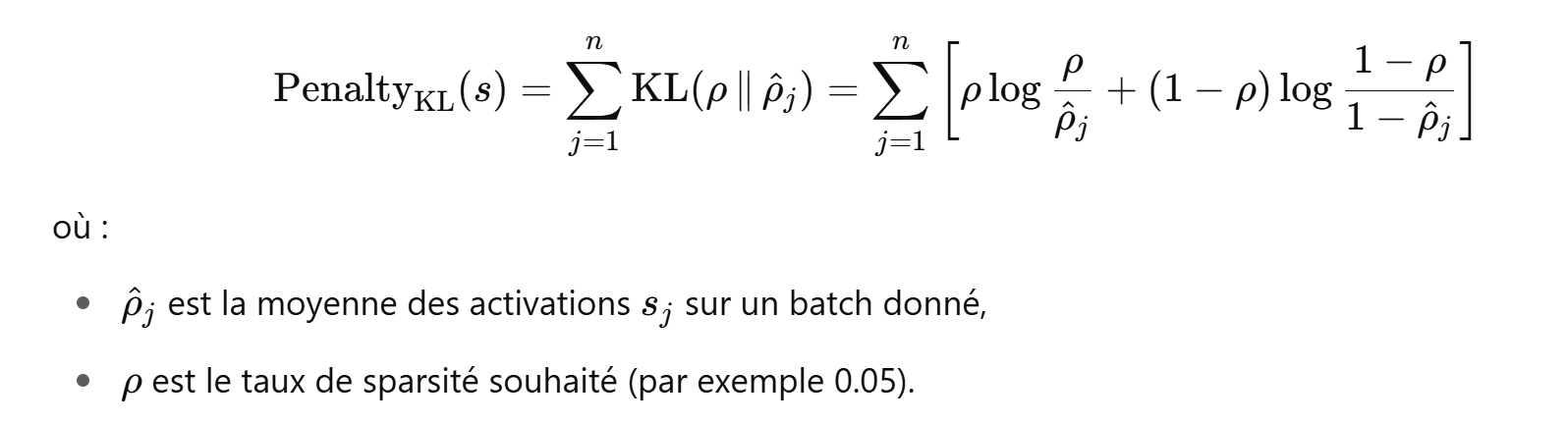
Cette pénalisation favorise l’annulation de la plupart des composantes de sss tout en permettant à certaines d’atteindre des valeurs significatives. Elle s’intègre naturellement dans les frameworks d’optimisation classiques et ne nécessite aucune hypothèse probabiliste sur la distribution des activations.

Le gradient de la norme , est constant, ce qui génère une pression régulière sur toutes les activations non nulles, favorisant ainsi leur réduction progressive vers zéro sans bloquer le gradient (contrairement à , non dérivable).

#### 2. Divergence de Kullback-Leibler (KL)

Une autre méthode repose sur la minimisation de la divergence entre la distribution réelle des activations et une distribution cible, généralement une Bernoulli avec une moyenne faible ρ≪1. Cette méthode est utilisée notamment dans les sparse autoencoders classiques.

La régularisation KL s’écrit :



Cette formulation impose que les activations latentes soient proches de zéro en moyenne, avec une interprétation probabiliste plus rigoureuse que le , mais au prix d’un calcul plus complexe.

#### Comparaison des méthodes

|  |  |  |
| --- | --- | --- |
| **Critère** |  | **KL-divergence** |
| Simplicité d’implémentation | Haute | Moyenne |
| Interprétation probabiliste | Non | Oui |
| Dérivabilité | Continue sauf en zéro | Continue |
| Sensibilité au taux de sparsité | Faible | Élevée (via ρ) |
| Adaptée à l’inférence embarquée | Oui | Moins directe |

Dans le cadre de notre projet, la régularisation L1L^1L1 a été privilégiée pour sa simplicité, sa stabilité numérique, et sa compatibilité directe avec les frameworks de compression pour microcontrôleurs. Elle est implémentée comme suit dans notre code expérimental :

def sparsity\_loss(latent):

return torch.mean(torch.abs(latent))

Cette fonction est ajoutée à la perte de reconstruction avec un coefficient λ déterminé empiriquement.