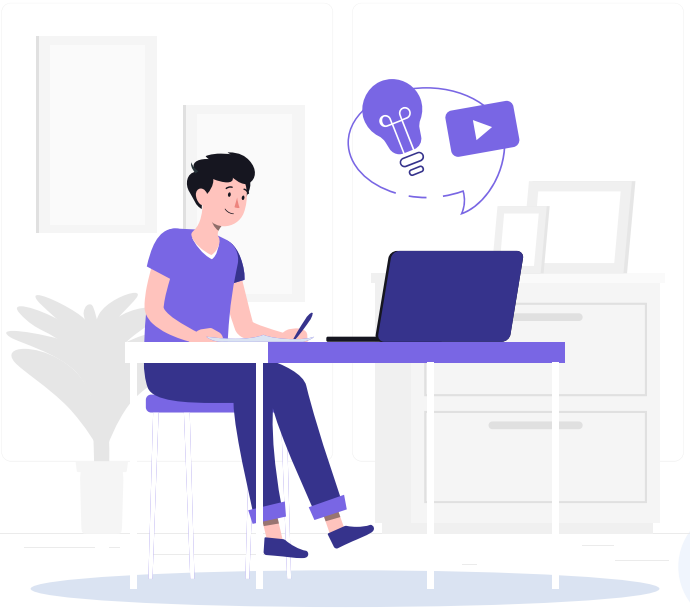
Deep Learning

Nissrine Hatibi



Classification des Séquences Promotrices du Génome Humain

Classification des Séquences Promotrices du Génome Humain

La transcription est le processus de copie du matériel génétique (ADN) en ARN.

Déterminer à quel endroit une région d'ADN doit commencer à être transcrite :

→ c'est le rôle du promoteur.

Classification des Séquences Promotrices du Génome Humain

Un promoteur, ou séquence promotrice, est une région de l'ADN située à proximité d'un gène, il est indispensable à la transcription de l'ADN en ARN.

C’est la zone sur laquelle se fixe initialement l'ARN polymérase, avant de démarrer la synthèse de l'ARN.

Transcription | Promoteur

Classification des séquences promotrices du génome humain suivant l'approche ULMFiT

ULMFiT - une méthode de Transfer Learning pour les tâches NLP.
Le Transfer Learning est le processus consistant à utiliser un modèle formé pour résoudre une tâche comme base pour résoudre une autre tâche. Cela signifie qu'au lieu de former un modèle à partir de zéro pour la deuxième tâche, on initialise le modèle avec les poids appris de la tâche initiale, puis on affine les poids vers la deuxième tâche.

Deux types d'ensembles de données sont utilisés dans le processus ULMFiT.

- Le premier est l'ensemble de données utilisé pour la formation du general domain genomic language model.
- Le second est l'ensemble de données utilisé pour la tâche de classification cible.

→ Pour créer des ensembles de données pour la formation du general domain genomic language model, les génomes complets sont téléchargés à partir du NCBI.

→ Les ensembles de données utilisés pour les tâches de classification sont extraits de publications existantes : « Recognition of Prokaryotic and Eukaryotic Promoters using Convolutional Deep Learning Neural Networks ».

Tokeniser par nucléotide

La tokenisation d'un seul nucléotide traiterait la séquence ATCGCGTACG en A T C G C G T A C G

→ Une représentation très restreinte des sous-unités génomiques.

Tokeniser par k-mers

Tokeniser la séquence ATCGCGTACGATCCG en:
* 3-mers: ATC GCG TAC GAT CCG
* 4-mers: ATCG CGTA CGAT
* 5-mers: ATCGC GTACG ATCCG
* Ou une autre taille k-mer.

Tokenisation de la séquence ATCGCGTACGATCCG avec une taille k-mer de 4 et les valeurs de stride :

- Stride 1: ATCG TCGC CGCG GCGT CGTA GTAC TACG ACGA CGAT GATC ATCC TCCG
- Stride 2: ATCG CGCG CGTA TACG CGAT ATCC
- Stride 3: ATCG GCGT TACG GATC
- Stride 4: ATCG CGTA CGAT

Créer un dictionnaire mappant chaque k-mer unique à une valeur entière.

Cela crée le vocabulaire du modèle.

→ K-mers = 3 ; Stride = 2

Séquence: ATCGCGTACGATCCG
Tokenisation: ATC CGC CGT TAC CGA ATC CCG
Numérisation: [5, 12, 8, 32, 27, 5, 14]

ULMFiT
Universal Language Model Fine-Tuning for Text Classification

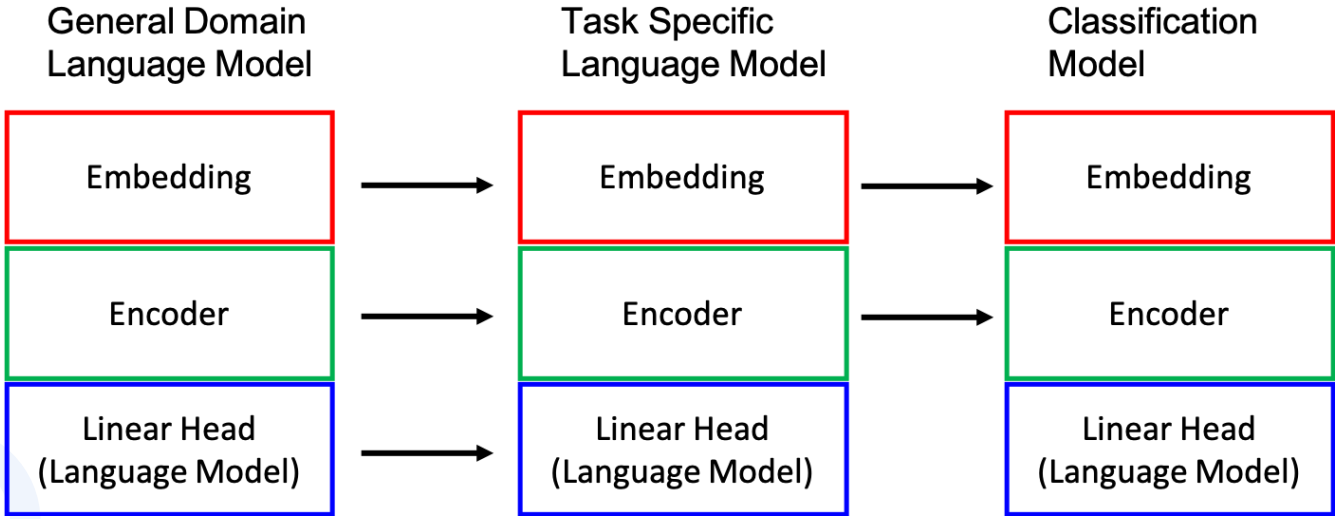
ULMFiT divise le training en 3 étapes :

1. Training d'un LM général.
2. Spécialisation par Transfer Learning du LM général pré-entrainé.
3. Training d'un classificateur dont les 1ères couches sont issues par Transfer Learning du LM spécialisé.

Un Language Model est un modèle qui prend une séquence de tokens k-mer et prédit le token suivant dans la séquence.

Un Classification Model est un modèle qui prend une séquence de tokens et prédit à quelle catégorie ou classe appartient cette séquence.

Les architectures du modèle de classification et du modèle de langage suivent des structures similaires - elles consistent en un Embedding, un Encoder, et un Linear Head



L'entrée ultime dans le modèle est un vecteur d'entiers.
Par exemple: [5, 12, 8, 32, 27, 5, 14]

La matrice de poids d'Embeddings aura une taille de vocab x n_embedding

Embeddings - size (vocab, n_embedding)

La section encodeur est composée de trois couches LSTM empilées.

Les couches LSTM sont structurées de telle sorte que le nombre d'unités cachées augmente, puis se contracte. Une structure standard serait:

```
LSTM(n_embedding, n_hidden)
LSTM(n_hidden, n_hidden)
LSTM(n_hidden, n_embedding)
```

Différentes linear heads sont utilisées pour le modèle de language et le modèle de classification, car chaque modèle effectue la classification à des fins différentes et utilise les hidden states de la final LSTM layer de différentes manières.

Language Model Head

Prédire le prochain k-mer dans une séquence génomique, produisant un vecteur de classification de longueur égale au vocabulaire du modèle.

Linear - size (400, vocab)

Classification Model Head

Effectuer des prédictions de classification sur le nombres de classes dans l'ensemble de données.

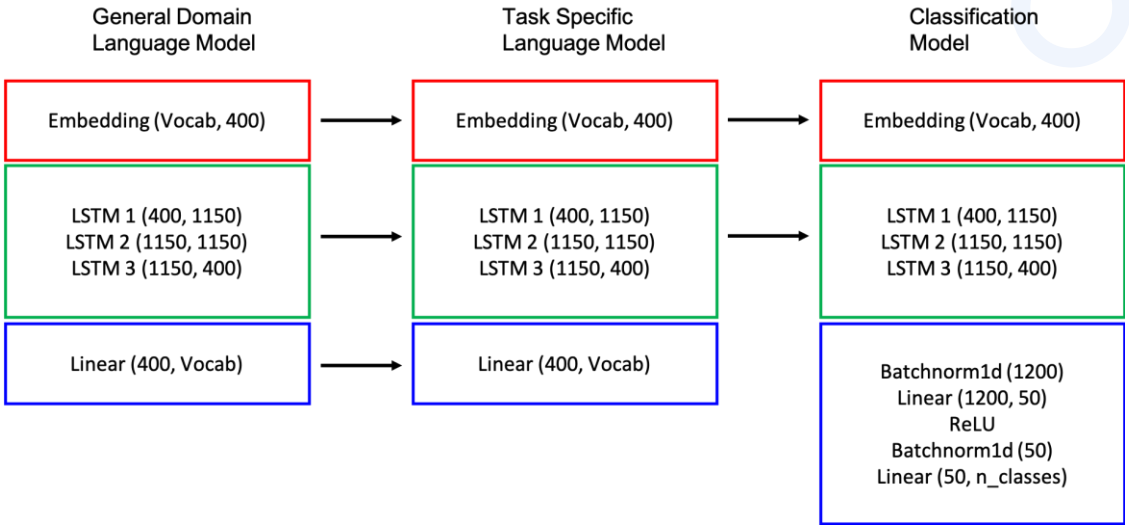
Batchnorm1d(n_embedding*3)
Linear(n_embedding*3, n) + bias
ReLU
Batchnorm1d(n)
Linear(n, n_classes) + bias

Embeddings
Embeddings - size (vocab, 400)

Encoder
LSTM 1 - size (400, 1150)
LSTM 2 - size (1150, 1150)
LSTM 3 - size (1150, 400)

For the Language Model Head
Linear - size (400, vocab)

For the Classification Model Head
Batchnorm1d 1 - size (1200)
Linear 1 - size (1200, 50)
Batchnorm1d 2 - size (50)
Linear 2 - size (50, n_classes)



Hyperparamètres
* Embeddings (**400**)
* 3 LSTM layers (**1150** hidden activations per layer)
* Backpropagation through time (**70**)
* Dropout
    - (**0,02 / 0,1** → Embeddings)
    - (**0,15 / 0,5** → LSTM hidden-to-hidden matrix)
    - (**0,1 / 0,2** → Activations of the LSTM layers)
    - (**0,25 / 0,4** → Output linear layers)
*Weight Decay coefficient (**0,01**)

Optimizer : AdamW – Coefficients : β1 = 0,9 / β2 = 0,99
Learning Rate : 5e – 4 → 1e – 2 (learning rates on the higher end of the range)

Ensemble de données

46 826 :

* 19 787 → Séquences promotrices
* 27 038 → Séquences régulatrices

	Sequence	Promoter	set
0	GGAGAGTTCCAGTCCCCCCAGCTGTCATGTTTAAAATCTGAAACAA...	1	train
1	AATTTATAGGCTAATGAGCAGAGACCCACCTTAACAAAATTTTGTC...	1	train
2	AGCAGGCTCAGAACTCGGAACAAAAGCAGTGACTGCATAATAATGC...	1	train
3	TGTGTCTCACGGATTCCACACCCGGCCGCTCCTGGACAAGCCCCGA...	1	train
4	GTCCCTGGGTGACCTTGCTGCCCTTGGAGAGCCATTCCAGGCAAAT...	1	train

Training	Validation	Test
35 821	3 979	7 025

Naive Model

Accuracy: 0.8704626334519573
False Positives: 0.031174377224199287
False Negatives: 0.09836298932384342
Recall: 0.7672617042775345
Precision: 0.9122947537044453
Specificity: 0.9460059171597633
MCC: 0.7361123456013181

Model with Genomic Pretraining

Accuracy: 0.9167259786476868
False Positives: 0.01807829181494662
False Negatives: 0.06519572953736655
Recall: 0.8457393061636915
Precision: 0.9518574677786201
Specificity: 0.9686883629191322
MCC: 0.8307787490889034

Model with Genomic Pretraining and Fine Tuning

Accuracy: 0.9738078291814947
False Positives: 0.015516014234875445
False Negatives: 0.010676156583629894
Recall: 0.9747389693499495
Precision: 0.9637029637029637
Specificity: 0.9731262327416174
MCC: 0.9464593207474515

Références

- https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0171410
- https://planet-vie.ens.fr/thematiques/cellules-et-molecules/physiologie-cellulaire/la-transcription-chez-les-eucaryotes
- https://www.ncbi.nlm.nih.gov/
- https://github.com/solovictor/CNNPromoterData
- https://fastai1.fast.ai/index.html
- https://pytorch.org/
- https://intelligence-artificielle.agency/notes-sur-la-lecon-1-de-fastai-deep-learning/