ALTeGraD Data Challenge

Final Presentation

Erwan Fagnou, Denis Duval

Team: Peptide Bond 007

January 2023

Table of contents

Table of Contents

Protein graph dataset

- 4888/1223 protein graphs in the training/test set
  $\rightarrow$ We took 500 proteins to make a validation set
- Task : Classify the protein graphs into 18 different classes
- Nodes as well as edges have features

Discrepancies in the dataset

Figure: Distribution of the classes in the dataset

Discrepancies in the dataset

	Min	0.25-quantile	Median	Mean	0.75-quantile	Max
Nodes	9	129	221	257	340.5	989
Edges	55	2187	3996	4721	6491	20417

Table: Statistics on the number of nodes and edges of the protein graphs.

Table of Contents

First approach

Graph convolutional networks on the default features from the dataset.

First approach

Graph convolutional networks on the default features from the dataset.

Fancier methods: Graph Attention Networks, Attentive FP, and DeeperGCN.
$\rightarrow$ No improvement

First approach

Graph convolutional networks on the default features from the dataset.

Fancier methods: Graph Attention Networks, Attentive FP, and DeeperGCN.
$\rightarrow$ No improvement

Same for the aggregation part: the mean is the best.

Adding extra features

- From the literature:
  - Local Degree Profile
  - Laplacian Positional Embedding

Adding extra features

- From the literature:
  - Local Degree Profile
  - Laplacian Positional Embedding
- Custom features:
  - Torsion angle
  - Distance to the center of mass
  - Position in the sequence

Adding extra features

- From the literature:
  - Local Degree Profile (5)
  - Laplacian Positional Embedding (3)
- Custom features:
  - Torsion angle (2)
  - Distance to the center of mass (1)
  - Position in the sequence (1)

Adding extra features

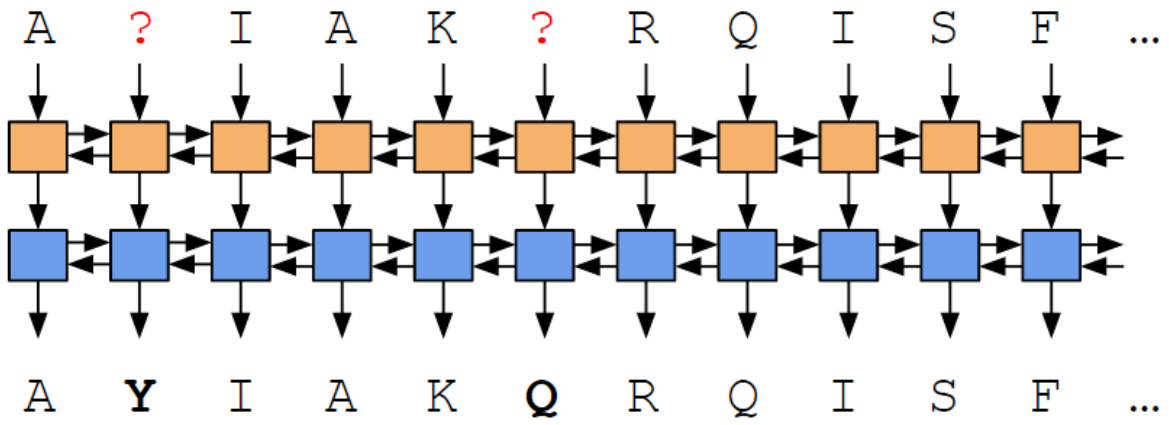
	Node features	
	default	default + extra features
Number of features	83	95
Validation loss	1.77	**1.62**

Table: Performance of the same GCN model using extra node features.

Table of Contents

Masked language modelling with Bidirectional LSTM

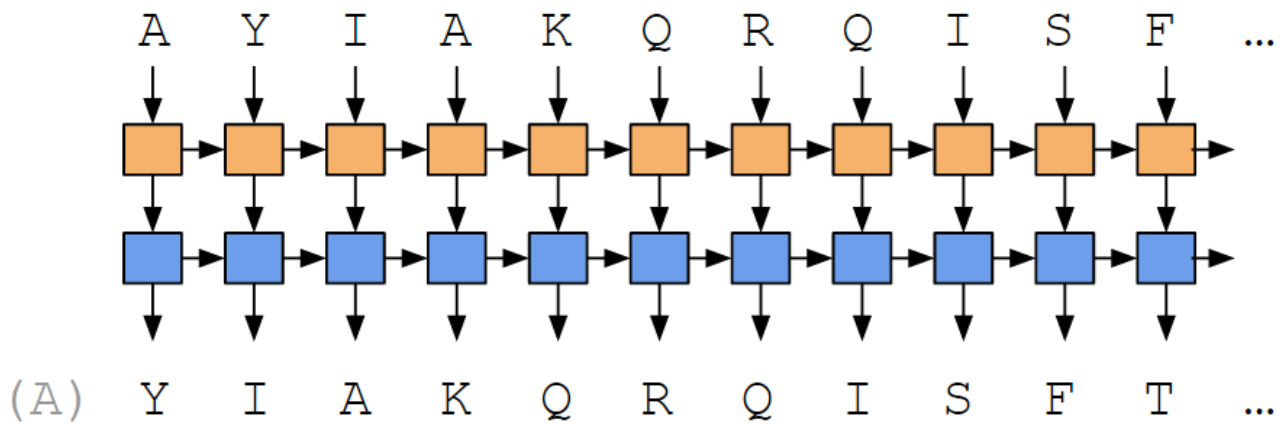
Pretraining task: masked language modelling



→ Quite bad...

Language modelling with LSTM

Pretraining task: language modelling



→ Already much better!

Performance with pretrained LSTM

Figure: Loss and accuracy during training with and without using a pretrained 2-layer LSTM for the node embeddings, before a GCN.

Combining different features

	Node features			
	default	default + extra features	LSTM	LSTM + extra features
Number of features	83	95	512	524
Validation loss	1.77	1.62	1.57	**1.49**

Table: Performance of the same GCN model using different node features. The default features from the dataset can be improved with additional features, while replacing them with a pretrained LSTM is even better.

Table of Contents

What is ESM-2?

- Transformer protein language model
  - Introduced by FAIR in 2019
  - Unsupervised pre-training on 250 millions protein sequences
- Main task is (atomic resolution) structure prediciton, SOTA model
- Trained with MLM objective

ESM-2 as a feature extractor

- One could just fine-tune the model with a classification head : quick overfitting
- Works better : train a model on the transformed dataset
  - Combinations of the hidden states as embeddings
  - Different sizes of ESM-2: 8M, 35M, 150M, 650M, 3B
- Different models :
  1. MLP
  2. GNN
  3. Multi Head Attention

Influence of the size of ESM-2

Figure: Loss and accuracy during training with respect to the size of ESM-2.

Table of Contents

Preventing overfitting

- Dropout (20%)

Preventing overfitting

- Dropout (20%)
- Label smoothing (0.05)

Label smoothing

Extreme case: output probabilities are 0 or 1 (before label smoothing)
Test accuracy $= 85\%$, label smoothing $= 0.05$, 18 classes

Label smoothing

Extreme case: output probabilities are 0 or 1 (before label smoothing)
Test accuracy $= 85\%$, label smoothing $= 0.05$, 18 classes

$$\Rightarrow \quad \mathcal{L}_{\text{test}} \approx -0.85 \log(1 - 0.05) - (1 - 0.85) \log \frac{0.05}{18 - 1}$$
$$\approx 0.92$$

Preventing overfitting

- Dropout (20%)
- Label smoothing (0.05)
- Reducing the embedding dimension (PCA or learned)

Preventing overfitting

- Dropout (20%)
- Label smoothing (0.05)
- Reducing the embedding dimension (PCA or learned)
- Tuning the learning rate

Tuning the learning rate

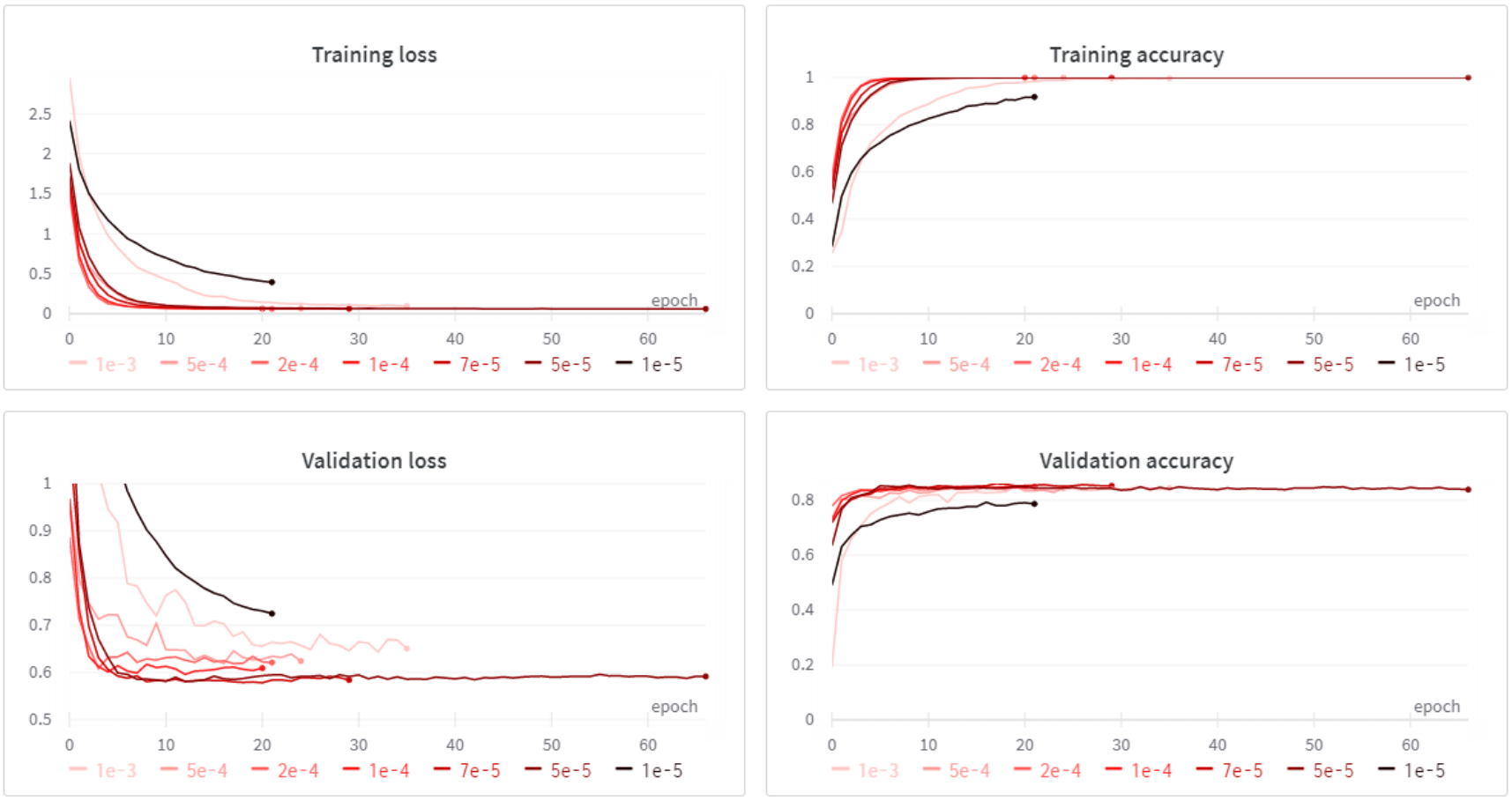


Figure: Loss and accuracy during training with respect to the learning rate, for train and validation data (same model, using ESM-2 3B).

Preventing overfitting

- Dropout (20%)
- Label smoothing (0.05)
- Reducing the embedding dimension (PCA or learned)
- Tuning the learning rate
- Averaging predictions of multiple models

Improvements with averaging

	Number of models			
	1	4	25	336
Average	0.684	0.638	0.628	**0.625**
Median	0.684	0.659	0.635	0.631

Table: Public score on Kaggle after aggregating the predictions of multiple models using ESM-2 650M, either with the average or the median.

Table of Contents

Scores of the models developed

	Sequence Baseline	GCN	GCN + LSTM	ESM-2 + MLP	ESM-2 + GNN	ESM-2 + MHA
Validation Loss	1.45	1.33	1.30	1.00	0.9	**0.4**
Validation Accuracy	$\lesssim 0.5$	0. 51	0.55	0.65	0.6	$\gtrsim$ **0.85**
Kaggle private score	1.68	1.51	1.55	1.18	1.21	**0.59**

Table: Best scores obtained for each method

- Best method : ESM-2 + MHA
- 1^{st} place on the leaderboard !

Tuning of the final solution : ESM-2 + MHA

ESM-2 embeddings with naive self attention layer is not enough. It scores only $\simeq 1.2$ of public score. In decreasing order, here is a list of hyper parameters with the greatest impact:

1. Size of the model : 3B
2. Number of models used for prediction (averaging) : 113
3. Combination of the hidden layers of ESM-2: (2, 13, 24, 35)
4. Label smoothing, dropout, initial learning rate: 0.5, 0.2, 7e-5
5. Hidden dimension, Number of queries, Number of heads : 128, 20, 4

Thank you!

Any question?