

Pruning Filters In Convolution Neural Network

Vincent Martineau

Department of Computer Science and Software Engineering, Université Laval



Introduction

We explore how reducing network expressivity can affect performance in Convolution Neural Network (CNN). We implemented a pruner that can remove filters from convolution layer and explore the effect on transfer learning tasks.

Motivations :

- **Reduce** network size.
- **Improve** execution speed.

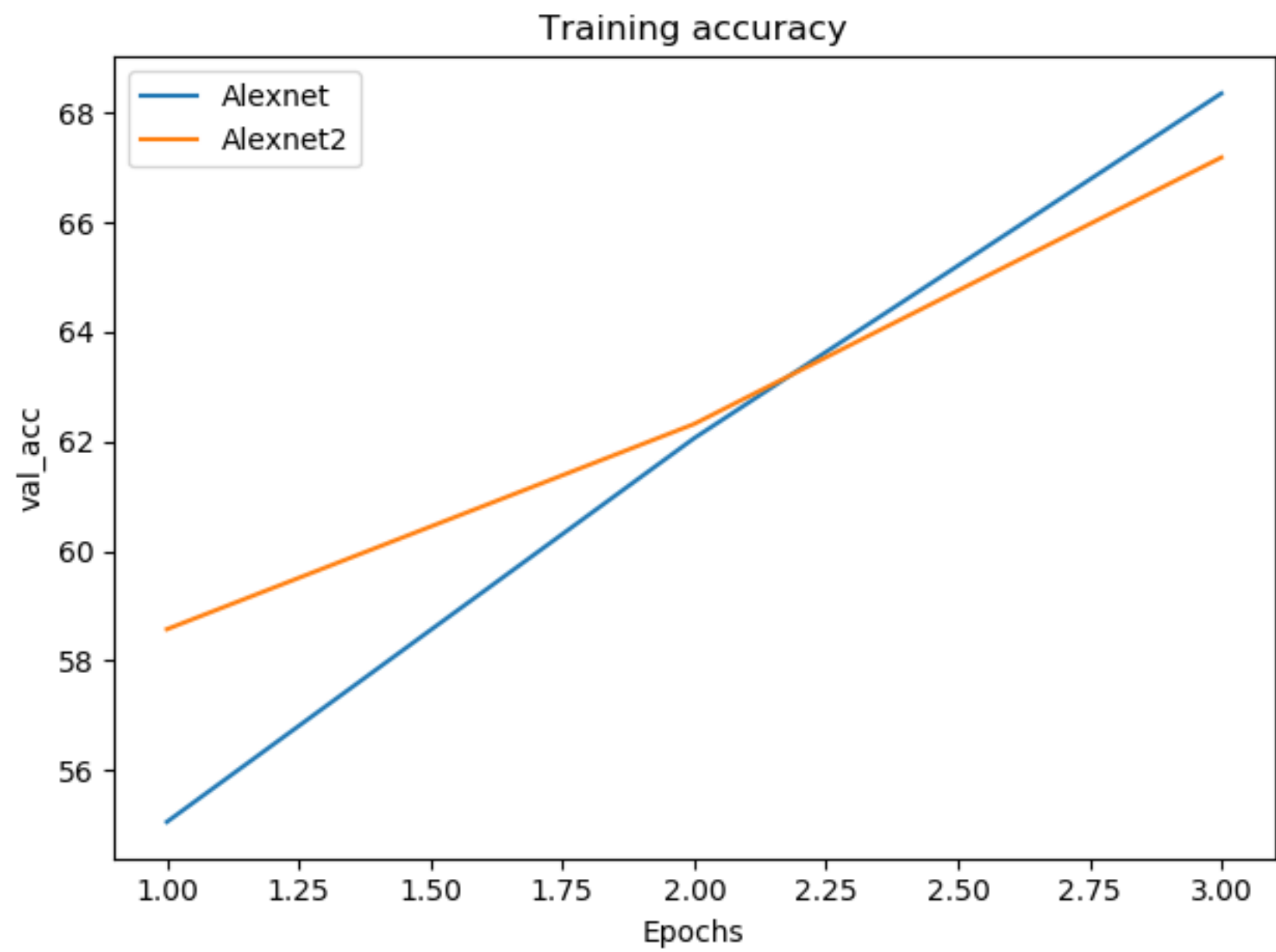
Related work :

- P.Molchanov et al. (2017) : Pruning Convolutional Neural Networks for Resource Efficient Inference.

Goals :

- Evaluate the impact of reducing network expression on performance.
- Compare training time on various model.
- Compare various strategies to prune.
- Provide a module that could.

Comparing Various level of Pruning in AlexNet



The net consists in 3 bi-LSTM taking as input the left context, the right context and the word characters. An attention module ponderates their outputs which are then combined in a last fully connected layer.

Experiments

Set up :

- Labeling tasks :
 - **Named Entity Recognition** (NER).
 - **POS tagging** (POS).
- Dataset : **CoNLL 2003**

Training details :

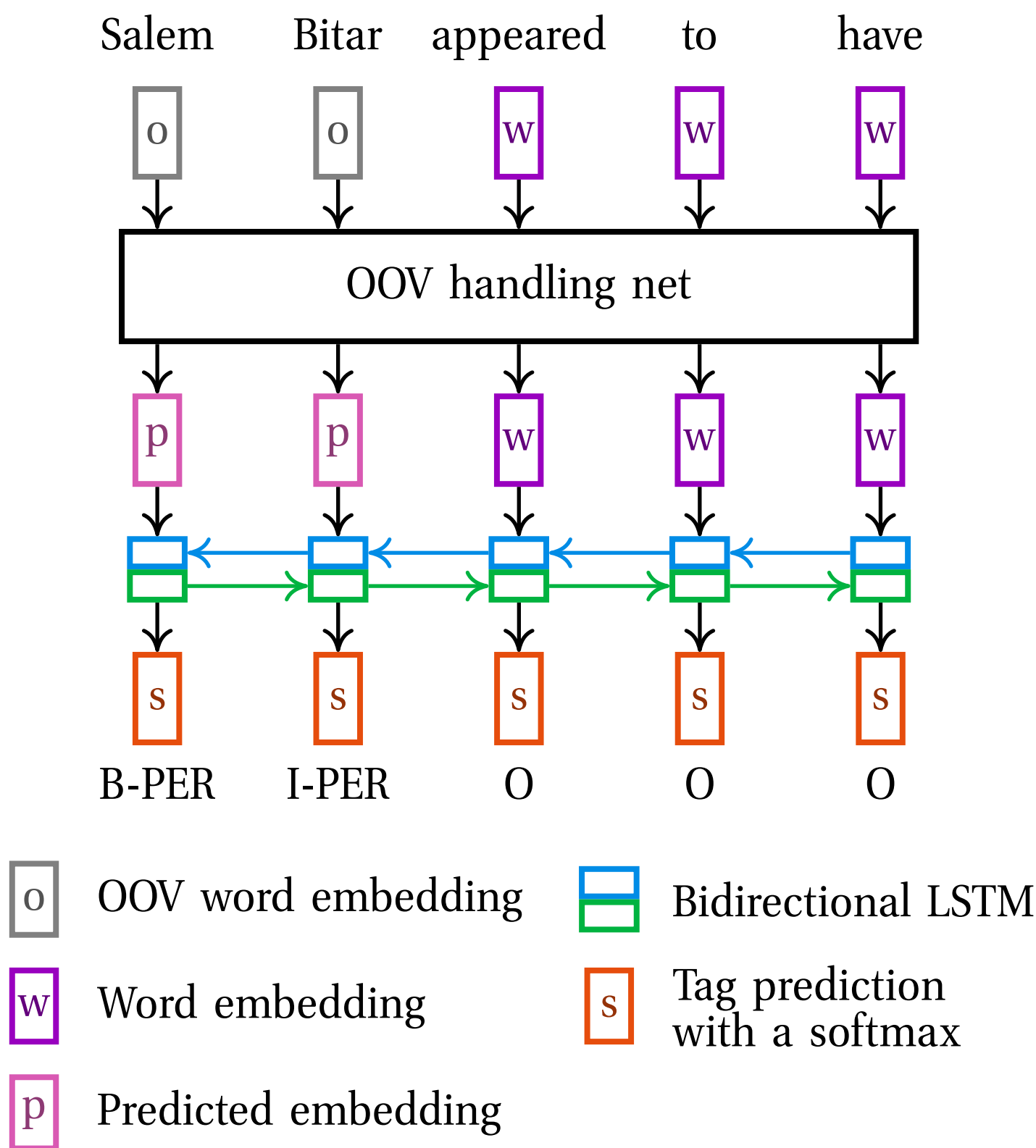
- Tensors sizes :
 - Char. emb. : 20.
 - Word emb. : 100 (**GloVe**).
 - LSTMs hidden state : 128.
- Context size from 2 words to the whole sentence.
- Standard learning rate on the labeling task parameters, reduced learning rate on Comick using SGD (0.01, 0.001).

Examples

Entity	Ponderation			
	Word	Left	Right	
PER	0.19	0.49	0.32	in sentencing darrel vo
PER	0.15	0.59	0.26	<BOS> australian parliamentarian
PER	0.15	0.61	0.24	had received today from mr j
PER	0.15	0.69	0.16	<BOS> rtrs - australian
ORG	0.22	0.46	0.32	the number of plastic surgeries in
ORG	0.28	0.23	0.49	to increase them in the united s
LOC	0.16	0.22	0.62	some residents of the <i>kazanlu</i>
LOC	0.20	0.47	0.33	at a mosque in the central bulgar
MISC	0.68	0.11	0.21	freestyle s
MISC	0.42	0.18	0.40	the <i>franco-african</i> summit

Qualitative example on several OOV words (underlined). We can see that depending on the context and the target, the weights may shift drastically.

Labeling task net



Two nets working together : the first predicts OOV embeddings (see OOV handling net section) and the second one predicts tags. The simple architecture of the labeling net is used to emphasize the usefulness of our module, and to minimize the influence of other factors.

Interpretability

Task	Tag	Ex.	Ponderation		
			Word	Left	Right
NER	O	1039	0.81	0.08	0.11
	B-PERS	63	0.21	0.31	0.49
	I-PER	119	0.16	0.52	0.32
	B-ORG	40	0.26	0.30	0.44
	I-ORG	3	0.27	0.31	0.42
	B-LOC	13	0.23	0.30	0.47
	I-LOC	2	0.16	0.48	0.36
	B-MISC	47	0.40	0.21	0.39
POS	I-MISC	5	0.41	0.26	0.33
	NNP	308	0.29	0.31	0.40
	NN	46	0.45	0.20	0.35
	CD	827	0.86	0.05	0.09
	NNS	23	0.37	0.24	0.39
	JJ	100	0.49	0.15	0.36

Average weights assigned to word's characters, left context and right context by the attention mechanism. We can clearly see the shift of attention according to the target entity. We also observe that the attention depends on the task at hand.

Performance gain

Task	Metric	Random Emb.	Our module	Gain
NER	F1	77.56	80.62	3.9%
POS	acc.	91.41	92.58	1.2%

The impact of our model on two NLP downstream tasks. We compare our OOV embeddings prediction scheme against random embeddings.

Conclusion

Discussion :

- **Morphology** and **context** help predict useful embeddings.
- **The attention mechanism works** : depending on the task, the network will use either more the context or the morphology to generate an embedding.

Future works :

- Apply the **attention mechanism on each character of the OOV word and each word of the context** instead of using the hidden state of the respective elements only.
- Test our attention model in **different languages** and on other NLP tasks, such as **machine translation**.