Identifying and Controlling Neurons in Neural Networks

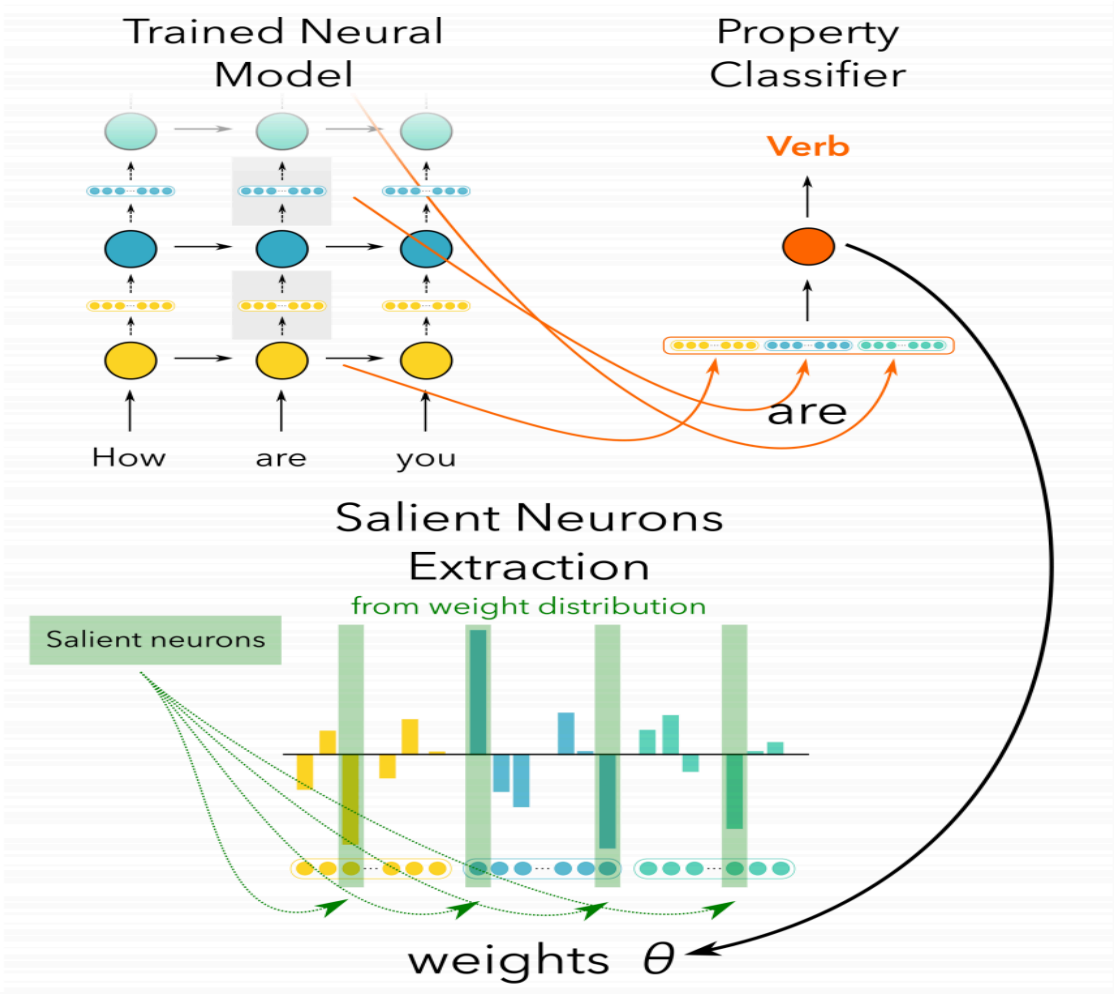
Jie Hao

- Analysis:
  - What Is One Grain of Sand in the Desert? Analyzing Individual Neurons in Deep NLP Models. (AAAI 2019)
  - Identifying and Controlling Important Neurons in Neural Machine Translation. (ICLR 2019)
  - How Important is a Neuron? (ICLR 2019)
  - On the Importance of Single Direction for Generalization. (ICLR 2018)
- Model Modification:
  - Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks. (ICLR 2019)
- Training Modification:
  - Neural Rejuvenation: Improving Deep Network Training by Enhancing Computational Resource Utilization. (CVPR 2019)

Incomplete statistics

Analysis: Reranking & Ordering

- Linguistic Correlation Analysis



[AAAI 2019]

- Cross-model Correlation Analysis: Pearson Correlation Coefficient

$$score(\mathbb{M}_{ij}) = \max_{\substack{1 \le i' \le N \\ 1 \le j' \le D \\ i \neq i'}} \rho(\mathbb{M}_{ij}, \mathbb{M}_{i'j'})$$

Where M_{ij} is the j-th neuron in the i-th model and ρ is the Pearson correlation corfficient.

[AAAI 2019]

Effect of neuron ablation on BLEU

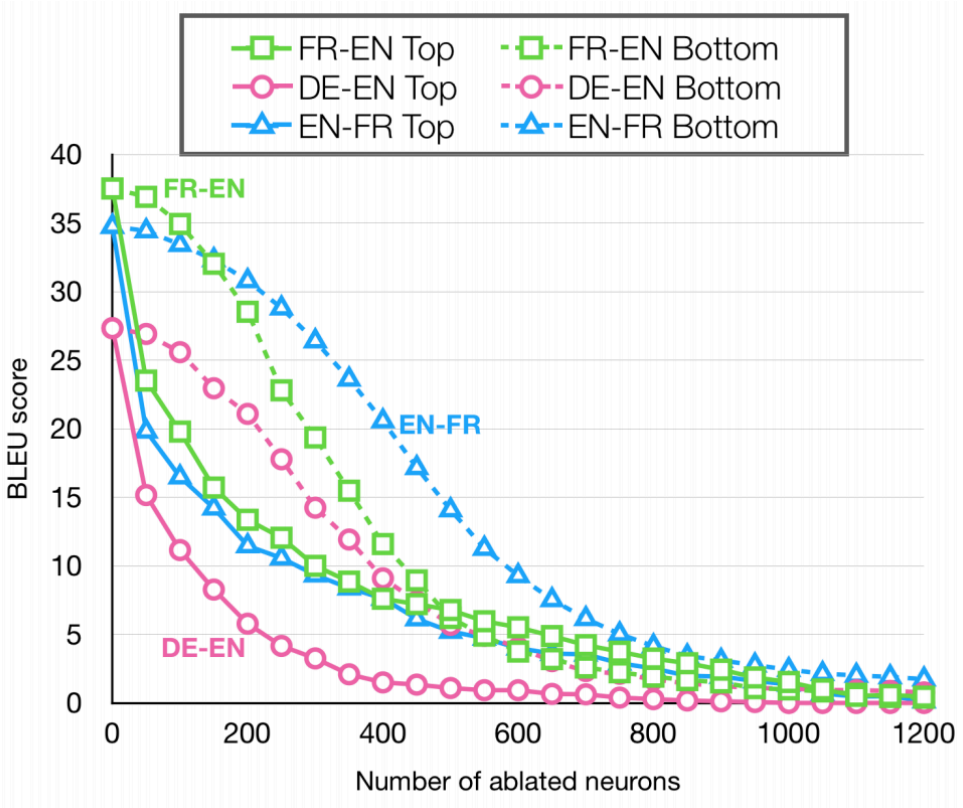


Figure 4: Effect of neuron ablation on translation performance (BLEU) when removing the top or bottom neurons based on Cross-Correlation analysis ordering.

Effect of neuron ablation on BLEU

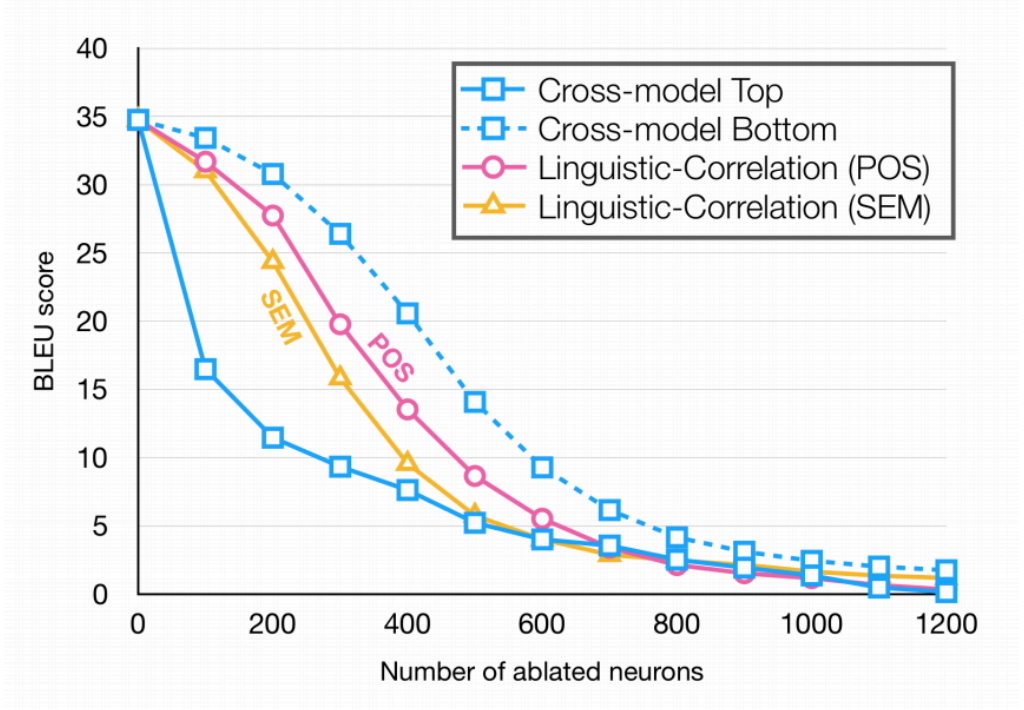


Figure 8: Effect on translation when ablating neurons in the order determined by both methods on the EN-FR model

- Cross-model Correlation Analysis

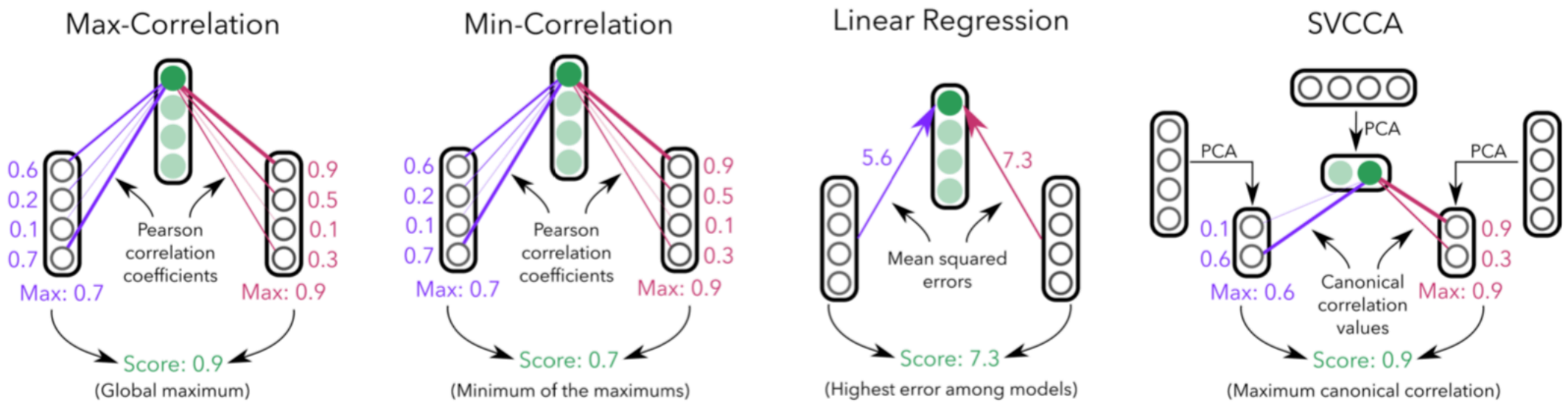


Figure 1: An illustration of the correlation methods, showing how to compute the score for one neuron using each of the methods. Here the number of models is $M = 3$, each having four neurons.

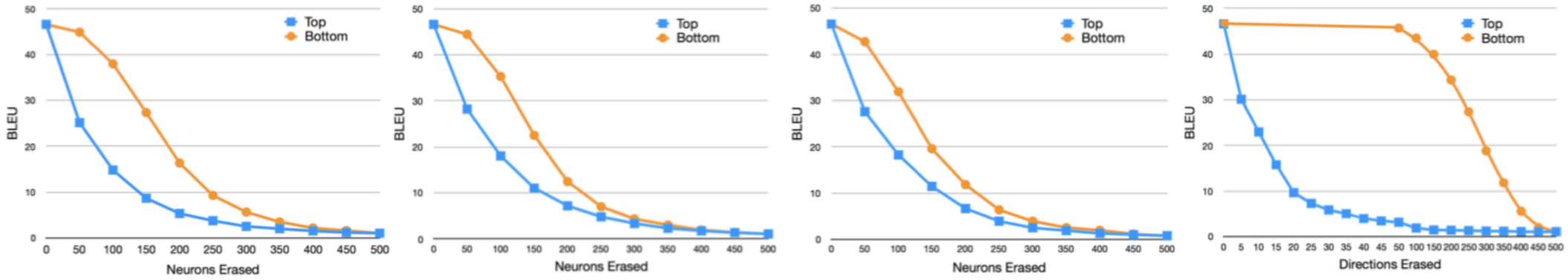
[ICLR 2019]

SVCCA: Singular Vector Canonical correlation Analysis. [NIPS 2017]

- **Input:** SVCCA takes as input two (not necessarily different) sets of neurons (typically layers of a network) $l_1 = \{\boldsymbol{z}_1^{l_1}, ..., \boldsymbol{z}_{m_1}^{l_1}\}$ and $l_2 = \{\boldsymbol{z}_1^{l_2}, ..., \boldsymbol{z}_{m_2}^{l_2}\}$
- **Step 1** First SVCCA performs a singular value decomposition of each subspace to get subsubspaces $l_1' \subset l_1, l_2' \subset l_2$ which comprise of the most important directions of the original subspaces l_1, l_2. In general we take enough directions to explain 99% of variance in the subspace. This is especially important in neural network representations, where as we will show many low variance directions (neurons) are primarily noise.
- **Step 2** Second, compute the Canonical Correlation similarity ([5]) of l_1', l_2': linearly transform l_1', l_2' to be as aligned as possible and compute correlation coefficients. In particular, given the output of step 1, $l_1' = \{\boldsymbol{z'}_1^{l_1}, ..., \boldsymbol{z'}_{m_1'}^{l_1}\}, l_2' = \{\boldsymbol{z'}_1^{l_2}, ..., \boldsymbol{z'}_{m_2'}^{l_2}\}$, CCA linearly transforms these subspaces $\tilde{l}_1 = W_X l_1'$, $\tilde{l}_2 = W_Y l_2'$ such as to maximize the correlations $corrs = \{\rho_1, \dots \rho_{\min(m_1', m_2')}\}$ between the transformed subspaces.
- **Output:** With these steps, SVCCA outputs pairs of aligned directions, $(\tilde{\boldsymbol{z}}_i^{l_1}, \tilde{\boldsymbol{z}}_i^{l_2})$ and how well they correlate, ρ_i. Step 1 also produces intermediate output in the form of the top singular values and directions.

Erasing Neurons

- Erasing neurons (or SVCCA directions) from the top and bottom of the list of most important neurons (directions) ranked by different unsupervised methods, in an English-Spanish model.



(a) MaxCorr (b) MinCorr (c) LinReg (d) SVCCA

Linguistic Properties

- Parentheses:

Figure 4: Visualization of a parentheses neuron from an English-Spanish model.

- Tense:

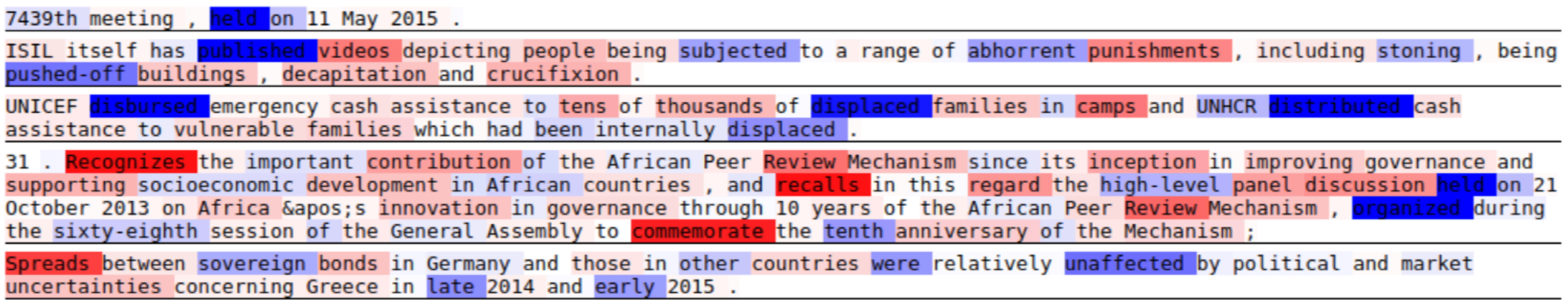
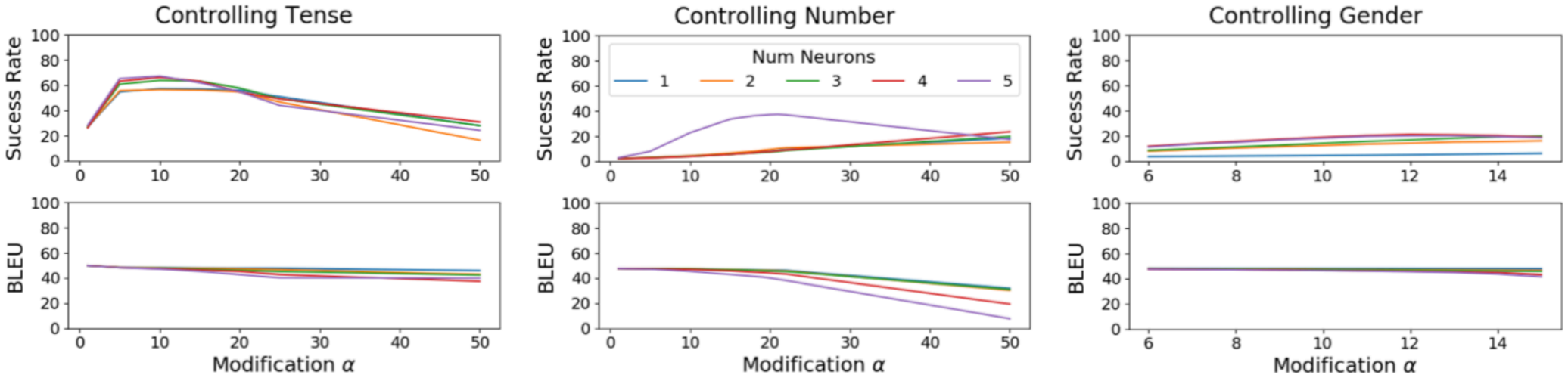


Figure 5: Visualization of a neuron from an English-Arabic model that activates on verb tense: negative/positive for past/present. Examples shown are the first 5 sentences in the test set.

Controlling neurons



(a) Tense (b) Number (c) Gender

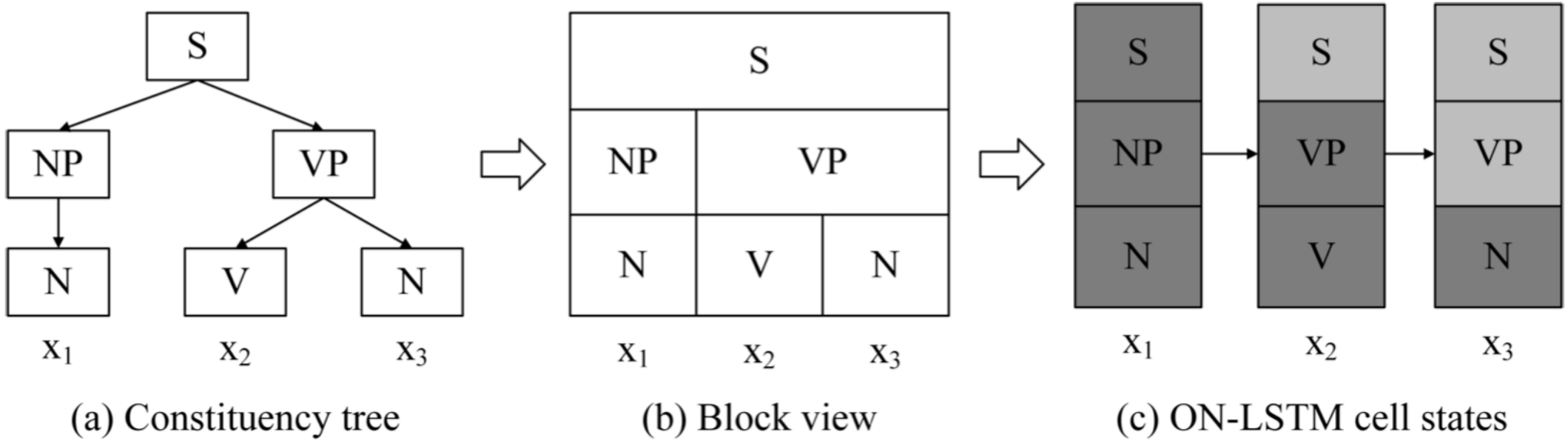
Figure 6: Success rates and BLEU scores for controlling NMT by modifying neuron activations.

Other Exploration:

- On the Importance of Single Direction for Generalization. (ICLR 2018)
  - Generalization capability is related to a network’ s reliance on single directions.
  - They also show that batch normalization, a highly successful regularizer, seems to implicitly discourage reliance on single directions.
  - Analysis on single dimension may be not a right way. Neural Network is still a blackbox!

Model Modification (ICLR 2019)

Inductive Bias --- Ordered Neurons



Training Modification (CVPR 2019)

- Neural Rejuvenation (NR): an optimization method for enhancing resource utilization during training.
  - Resource utilization monitoring.
  - Dead neuron rejuvenation.
  - Training with mixed neural types
- Let $\mathrm{C}(A)$ denote the cost of using architecture A, e.g. it can be the number of parameters or FLOPs. $U(A)$ denotes actual utilization of the computational resource of A.
- Utilization ratio: $R(A) = U(A) \,/\, \mathrm{C}(A)$.
- Goal: move $U(A)$ towards $\mathrm{C}(A)$.

Framework

Algorithm 1: SGD with Neural Rejuvenation

Input : Learning rate ϵ, utilization threshold T_r, initial architecture $\mathcal{A}$ and $\theta_{\mathcal{A}}$, and resource constraint $\mathcal{C}$

while *stopping criterion not met*:
  Sample a minibatch $\{(x_1, y_1), ..., (x_m, y_m)\}$;
  Compute gradient $g \leftarrow \frac{1}{m} \nabla \sum_i \mathcal{L}(f(x_i; \mathcal{A}, \theta_{\mathcal{A}}), y_i)$;
  Apply update $\theta_{\mathcal{A}} = \theta_{\mathcal{A}} - \epsilon \cdot g$;
  if *neural rejuvenation flag is on*:
    Compute utilization ratio $r(\theta_{\mathcal{A}})$;
    if $r(\theta_{\mathcal{A}}) < T_r$:
      Rejuvenate dead neurons and obtain new $\mathcal{A}$ and $\theta_{\mathcal{A}}$ under resource constraint $\mathcal{C}$;
return Architecture $\mathcal{A}$ and its parameter $\theta_{\mathcal{A}}$;

Resource Utilization Monitoring

- Liveliness of Neurons

$$v_i = \gamma \cdot \frac{u_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \beta, \ \ \forall i \in \{1, ..., m\}$$

$$\text{where } \mu_{\mathcal{B}} = \frac{1}{m}\sum_{i=1}^{m} u_i \text{ and } \sigma_{\mathcal{B}}^2 = \frac{1}{m}\sum_{i=1}^{m}(u_i - \mu_{\mathcal{B}})^2$$

- A neuron is considered dead if the scaling parameter $\gamma < 0.01 \times \gamma_{max}$, where γ_{max} is the maximum γ in the same batch-normalized layer.
- Compute utilization ratio: via binary mask.

Dead Neuron Rejuvenation

- Resource Reallocation: iterative squeeze-and-expand algorithm. [MorphNet, NIPS 2017]
- Parameter Reinitialization: randomly initialized the parameters for the reallocation neurons.
- Neural Rescaling: rescale all neurons to the initial level.

Training with Mixed Types of Neurons

- S (survived) neurons and R (rejuvenated) neurons.

$$\mathcal{S}_{\text{out}} = W_{\mathcal{S}\to\mathcal{S}} * \mathcal{S}_{\text{in}} + W_{\mathcal{R}\to\mathcal{S}} * \mathcal{R}_{\text{in}}$$
$$\mathcal{R}_{\text{out}} = W_{\mathcal{S}\to\mathcal{R}} * \mathcal{S}_{\text{in}} + W_{\mathcal{R}\to\mathcal{R}} * \mathcal{R}_{\text{in}}$$

- When S does not need R: cross-attention between S and R

$$\mathcal{S}_{\text{out}} = W_{\mathcal{S}\to\mathcal{S}} * \mathcal{S}_{\text{in}} + 2 \cdot \sigma(W_{\mathcal{S}\to\mathcal{S}} * \mathcal{S}_{\text{in}}) W_{\mathcal{R}\to\mathcal{S}} * \mathcal{R}_{\text{in}}$$

$$\mathcal{R}_{\text{out}} = W_{\mathcal{R}\to\mathcal{R}} * \mathcal{R}_{\text{in}} + 2 \cdot \sigma(W_{\mathcal{R}\to\mathcal{R}} * \mathcal{R}_{\text{in}}) W_{\mathcal{S}\to\mathcal{R}} * \mathcal{S}_{\text{in}}$$

Experiments

Architecture	Baseline		Network Slimming [39]		Neural Rejuvenation	
	C10 (Params)	C100 (Params)	C10 (Params)	C100 (Params)	C10 (Params)	C100 (Params)
VGG-19 [53]	5.44 (20.04M)	23.11 (20.08M)	5.06 (10.07M)	24.92 (10.32M)	4.19 (9.99M)	21.53 (10.04M)
ResNet-164 [22]	6.11 (1.70M)	28.86 (1.73M)	5.65 (0.94M)	25.61 (0.96M)	5.13 (0.88M)	23.84 (0.92M)
DenseNet-100-40 [27]	3.64 (8.27M)	19.85 (8.37M)	3.75 (4.36M)	19.29 (4.65M)	3.40 (4.12M)	18.59 (4.31M)

Table 4. Neural Rejuvenation for model compression on CIFAR [32]. In the experiments for ImageNet, the computational resources are kept when rejuvenating dead neurons. But here, we set the resource target of neural rejuvenation to the half of the original usage. Then, our Neural Rejuvenation becomes a model compressing method, and thus can be compared with the state-of-the-art pruning method [39].

Conclusion

- Neuron level research is popular in recent years.
- Interpretable: detect important neurons (corresponding to linguistic properties), further control the neurons.
- Modification: Model or Training. (enhance certain properties modeling on neuron-level operation)