International Journal of Innovative
Computing, Information and Control
Volume **10**, Number **6**, December **2014**

ICIC International ©2014 ISSN 1349-4198
pp. 1969–1981

# LANGUAGE MODELING USING AUGMENTED ECHO STATE NETWORKS

Arnaud Rachez and Masafumi Hagiwara

School for Open and Environmental Systems
Faculty of Science and Technology
Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama-shi 223-8522, Japan
arnaud.rachez@gmail.com; hagiwara@z7.keio.ac.jp

Received December 2013; revised April 2014

Abstract. *Interest in natural language modeling using neural networks has been growing in the past decade. The objective of this paper is to investigate the predictive capabilities of echo state networks (ESNs) at the task of modeling English sentences. Based on the finding that ESNs exhibit a Markovian organization of their state space that makes them close to the widely used n-gram models, we describe two modifications of the conventional architecture that allow significant improvement by leveraging the kind of representation developed in the reservoir. Firstly, the addition of pre-recurrent features is shown to capture syntactic similarities between words and can be trained efficiently by using the contracting property of the reservoir to truncate the gradient descent. Secondly, the addition of multiple linear readouts using the mixture of experts framework is also shown to greatly improve accuracy while being trainable in parallel using Expectation-Maximization. Furthermore it can easily be transformed into a supervised mixture of expert model with several variations allowing reducing the training time and can take into account handmade features.*

**Keywords:** Language model, Recurrent neural network, Gradient descent, Expectation-maximisation, Multiple readout, Echo state

1. **Introduction.** Simple models such as the $n$-gram model can be trained efficiently using smoothing techniques [4] and allow quick building of an accurate language model that can then be used as a sub-part in a more complex system. However, these simple language models are criticized for they only capture superficial linguistic structures and are unable to take into account long term dependencies that occur in natural languages. Moreover, $n$-gram models suffer from the curse of dimensionality: for a vocabulary containing $|V|$ words, there are $|V|^n$ possible different sequences of words and thus $n$-grams require $|V|^n$ parameters. Since most of the possible combinations of words are never seen during the training phase, the parameters cannot be estimated properly using maximum likelihood. Although different smoothing techniques allow facing this data sparsity, they usually rely on simple notions of similarity to generalize. For example back-off $n$-gram models rely on the frequency of shorter sequences as described in [10].

Neural language models have been proposed to palliate some of these shortcomings.

- In [1, 5], time-delay neural networks are fed with a vector of $k$ concatenated words. These models are close to $n$-grams since they take into account only a finite context but can develop very complex distributed representations and automatically perform smoothing.

- Recurrent neural networks used in [6, 11, 12] can take an arbitrarily long context into account when predicting the next word and perform at the state of the art level in word recognition in speech processing [11].
- Recursive neural networks learn compositional representation of words and phrases by applying recursively the same neural network and obtain excellent results on several natural language processing tasks despite having been introduced very recently [15, 16].

Echo state networks are a rather recent development in the field of recurrent neural networks belonging to a collection of techniques called Reservoir Computing [19]. The philosophy adopted in Reservoir Computing is to consider the recurrent layer as a large reservoir of nonlinear transformations of the input data and decouple the learning of parameters inside and outside the reservoir. In echo state networks, since all the connections but those from the recurrent layer to the output units are fixed, learning is easy and can be realized only by inverting the design matrix. Echo state networks have already shown promising performance on time series prediction tasks but have seldom been used in more abstract settings such as natural language modeling. However, the contracting dynamic of their recurrent layer implies that they construct a representation of the input that is Markovian as was already explained for a broader class of networks in [17] and closely mirror $n$-gram clustering [14]. Furthermore, [18] showed that ESNs are able to perform at a level of accuracy similar to that of recurrent neural networks although they require a number of recurrent units far superior.

Starting from the results of [18], the objective of this paper is not only to improve the accuracy of language models using echo state networks but also to reduce the number of recurrent units used in the reservoir. Moreover we try to keep a reasonable training time and construct distributed representation of linguistic features. Based on the observation that the organization of the reservoir is Markovian we describe two modifications of conventional Echo State Networks. Firstly, the introduction of a pre-recurrent feature layer is shown to improve the accuracy of the language model and allow to re-organize the activation pattern in the recurrent layer. To learn the features, we justify the truncation of the back-propagation through time algorithm using the echo state property of the network and provide a rigorous mathematical analysis of the effect of the echo state property on the decay of the gradient. Secondly, using the mixture of experts paradigm to add multiple readouts to the network is also shown to increase the performance while the training time remains acceptable when Expectation-Maximization is used to train the experts in parallel. Finally a simplification of the mixture of experts paradigm where the gating units are observed is investigated and its relation with the hierarchical probabilistic language model of [13] as well as other variations are discussed. As discussed in the last part of the paper, the models investigated here can serve as a basis for the elaboration of more complicated models.

## 2. Conventional Echo State Networks.

Here we describe the traditional implementation of Echo State Networks (ESNs) used in [18]. This model serves as a baseline model against which we compare our improvements. As described in [8] ESNs are simple, discrete-time recurrent neural networks with three layers:

- an input layer $\mathbf{u} \in \mathbb{R}^I$,
- a recurrent layer $\mathbf{x} \in \mathbb{R}^D$ also called reservoir,
- an output layer $\mathbf{y} \in \mathbb{R}^O$ extracting information from the reservoir.

The core component of an Echo State Network is the reservoir: a large collection of randomly connected units, each of which computes a nonlinear transformation of the

input signal. A readout function is then used to extract the information contained in the representation of the input sequence constructed by the reservoir. The dynamic of the network is described by the following system of equations:

$$\mathbf{x}^t = E(\mathbf{u}^0, \dots, \mathbf{u}^t) \tag{1}$$
$$= g(\mathbf{W}^{in}\mathbf{u}^t + \mathbf{W}\mathbf{x}^{t-1}) \tag{2}$$
$$\mathbf{y}^t = h(\mathbf{x}^t), \tag{3}$$
$$= \mathbf{W}^{out}\mathbf{x}^t, \tag{4}$$

where $g = \tanh$ and $h$ are respectively the activation of the recurrent layer and the readout function. $\mathbf{W}^{in} \in \mathbb{R}^{D\times I}$, $\mathbf{W} \in \mathbb{R}^{D\times D}$, $\mathbf{W}^{out} \in \mathbb{R}^{O\times D}$ are respectively the input, reservoir and output weight matrices. $E$ is the echo state function. It is to be noted that the echo state function has a growing number of input variable. $E(\mathbf{u}^0, \dots, \mathbf{u}^t)$ is actually an alternative way of expressing $\mathbf{x}^t$ (the activation of the reservoir at time $t$) making explicit the dependency on the input sequence $(\mathbf{u}^0, \dots, \mathbf{u}^t)$.

In order to keep the complexity of the system minimal, we chose not to admit back-propagation of the output units to the recurrent layer nor direct connection of the input units to the output layer. In a sense, this allows the network to be closer to the well known feed-forward architecture, a characteristic that is used later.

All the weight matrices $\mathbf{W}^{in}$, $\mathbf{W}$, $\mathbf{W}^{out}$ are usually initialized randomly and are kept fixed except for the output matrix which is learned. A sufficient condition on the reservoir weights matrix $W$ that ensures the Echo State Property is that its largest singular value $\bar{\sigma}(W)$ should be less than 1. In practice, although it does not always ensure the echo state property, the necessary condition on the reservoir weight matrix, i.e., $\rho(\mathbf{W}) < 1$, is commonly used as a good heuristic to construct the reservoir.

Learning the parameters by minimizing the loss function in the linear readout case is relatively easy. Denoting by $\mathbf{X}$ and $\mathbf{Y}$ the row matrices of (respectively) input vectore and output vectors are:

$$\mathbf{W}^{out} = \arg\max_{\mathbf{W}} L(\mathbf{Y}, h(\mathbf{X})) \tag{5}$$
$$= \arg\max_{\mathbf{W}} (\mathbf{Y} - \mathbf{W}\mathbf{X})^T(\mathbf{Y} - \mathbf{W}\mathbf{X}) \tag{6}$$
$$= \mathbf{Y}^T\mathbf{X}(\mathbf{X}\mathbf{X}^T)^{-1}, \tag{7}$$

where $\mathbf{X}^T$ denotes the transposition of $\mathbf{X}$.

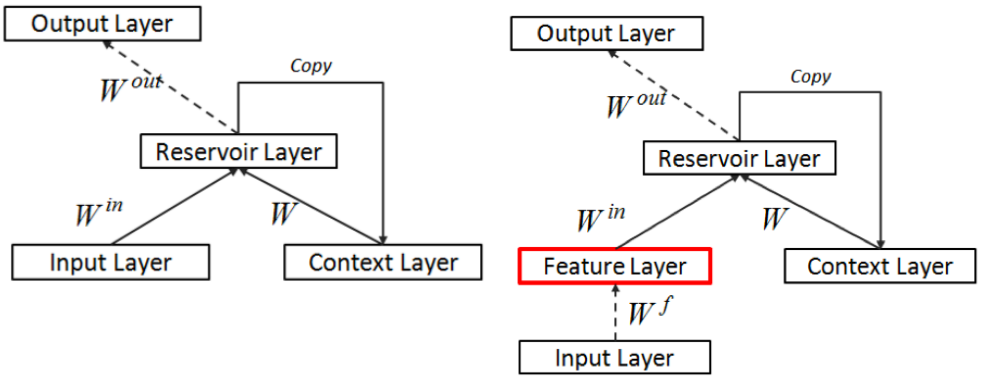
## 3. Augmented Echo State Network.

In this section, we propose two modifications to the traditional ESNs that allow improving their modeling power while capturing interesting syntactic and semantic properties. We also introduce the techniques used to perform learning with these modifications and keep the training time acceptabe. The main result of this section is a rigorous analysis of the impact of the echo state property on the decay of the gradient through time that is dual to the Markovian organisation of the reservoir space [14]. More sophisticated readout function and their training are also investigated.

### 3.1. Pre-recurrent features.

The system of equations describing the network augmented with a feature layer shown in Figure 1 is:

$$\mathbf{x}^t = E(f(\mathbf{u}^0), \dots, \mathbf{f}(\mathbf{u}^t)) \tag{8}$$
$$= g(f(\mathbf{u}^t), \mathbf{x}^{t-1}) \tag{9}$$
$$\mathbf{y}^t = h(\mathbf{x}^t), \tag{10}$$




(a) Conventional ESN

(b) Proposed network with a pre-recurrent feature layer

FIGURE 1. Comparison of the two architectures. Only the weights corresponding to dashed arrows are learned.

where $f$ is the function extracting features from the input $\mathbf{u}$. Following the idea presented in [1] and also used in [5], $f$ is a linear projection of the input vector $\mathbf{u}$ on a vector space whose dimensionality $F$ can be chosen as:

$$f(\mathbf{u}) = \mathbf{W}^f \mathbf{u}, \tag{11}$$

where $\mathbf{W}^f \in \mathbb{R}^{I \times F}$. Since the input vectors $\mathbf{u}$ are one-hot representations of words, the feature of a word is just the recopy of one column of the feature matrix into the feature layer.

3.2. **Learning with gradient descent.** The features matrix $\mathbf{W}^f$ and output matrix $\mathbf{W}^{out}$ are learned by minimizing the squared sum of error function $C$ of every sequence $s = (\mathbf{u}^0, \dots, \mathbf{u}^{T_s})$ in the training set $S$. At each time step the prediction of the network $\mathbf{y}_s^t$ is compared to the actual next word in the sequence $\mathbf{u}_s^{t+1}$,

$$C = \frac{1}{2} \sum_{s \in S} \sum_{t=1}^{T_s - 1} (\mathbf{y}_s^t - \mathbf{u}_s^{t+1})^T (\mathbf{y}_s^t - \mathbf{u}_s^{t+1}). \tag{12}$$

When learning the matrix $\mathbf{W}^{out}$, since all the parameters are situated after the recurrence, it is possible to use the least mean square algorithm to perform one shot learning. For the feature matrix $\mathbf{W}^f$, however, back-propagation through time must be used. Since the echo state property is responsible for the vanishing gradient effect described in [2], it is possible to discard part of the gradient while retaining a good approximation. Informally, since the influence of the feature $f(\mathbf{u}^t)$ is attenuated due to the contracting behaviour of the network, it has only a very limited impact on the state of the recurrent layer $\mathbf{x}^{t+n}$ if $n$ is a sufficient time difference. The update of the feature representation $f(\mathbf{u}^t)$ conditioned on the prediction of $\mathbf{u}^{t+1}$ can thus be several orders of magnitude bigger than the update caused by the prediction of $\mathbf{u}^{t+n}$.

In order to make the derivation of the learning procedure clearer, we briefly derive the matrix formulation of back-propagation through time. We consider that the Echo State Network fed with a sequence of length $L$ $\bar{\mathbf{u}} = \mathbf{u}^1, \dots, \mathbf{u}^L$ is unrolled in time. It has a structure equivalent to a feed-forward network with shared parameters at every instance $\mathbf{x}^t$ of the recurrent layer at time $t$. The system of equations describing the unrolled echo

state network is:

$$\mathbf{f}^t = \mathbf{W}^f \mathbf{u}^t \tag{13}$$

$$\mathbf{a}^t = \mathbf{W}\mathbf{x}^{t-1} + \mathbf{W}^{in}\mathbf{f}^t, \quad \text{for } 1 \leq t < L \tag{14}$$

$$\mathbf{x}^t = g(\mathbf{a}^t) \tag{15}$$

$$\mathbf{y}^t = \mathbf{W}^{out}\mathbf{x}^{t-1} \tag{16}$$

We are interested in the update of the feature $\mathbf{f}^1$ caused by the prediction error $e = (\mathbf{y}^{L-1} - \mathbf{u}^L)^T(\mathbf{y}^{L-1} - \mathbf{u}^L)$ at time $L-1$ that corresponds to only one term in the cost function of (12). The gradient of $e$ with respect to $\mathbf{f}$

$$\frac{\partial e}{\partial f_k^f} = \sum_{i,j} \frac{\partial e}{\partial a_i^1} \frac{\partial a_i^1}{\partial f_k^1} \tag{17}$$

$$= \sum_i \frac{\partial e}{\partial a_i^1} W_{i,k}^{in}, \tag{18}$$

can be expressed in matrix form:

$$\nabla_{\mathbf{f}} e = (\mathbf{W}^{in})^T \boldsymbol{\delta}^1 \tag{19}$$

where $\boldsymbol{\delta}^l = \frac{\partial e}{\partial \mathbf{a}^l}$. Moreover the error $\boldsymbol{\delta}^l$ at layer (or time) $l$ can be expressed relatively to $\boldsymbol{\delta}^{l+1}$ for $1 \leq l < L$:

$$\delta_p^l = \frac{\partial e}{\partial a^l} \tag{20}$$

$$= \sum_{i,j} \frac{\partial e}{\partial a_i^{l+1}} \frac{\partial a_i^{l+1}}{\partial x_j^l} \frac{\partial x_j^l}{\partial a_k^l} \tag{21}$$

$$= \sum_i \delta_i^{l+1} W_{i,p} g'(a_p^l), \tag{22}$$

or in matrix form:

$$\boldsymbol{\delta}^l = \mathbf{W}^T \boldsymbol{\delta}^{l+1} \cdot g'(\mathbf{a}^l) \tag{23}$$

Finally $\boldsymbol{\delta}^{L-1}$ at the last layer is:

$$\boldsymbol{\delta}^{L-1} = \frac{\partial e}{\partial \mathbf{a}^L} = \frac{\partial \frac{1}{2}(\mathbf{y}^{L-1} - \mathbf{u}^L)^T(\mathbf{y}^{L-1} - \mathbf{u}^L)}{\partial \mathbf{a}^L} = (\mathbf{W}^{out})^T \boldsymbol{\delta}^L \tag{24}$$

where $\boldsymbol{\delta}^L = (\mathbf{y}^{L-1} - \mathbf{u}^L)$. Using (19), (23) and (24) we can recursively calculate the update of $\mathbf{f}^1$. However, first let us see the form of the error $\boldsymbol{\delta}^1$:

$$\boldsymbol{\delta}^1 = \mathbf{W}^T \boldsymbol{\delta}^2 \cdot g'(\mathbf{a}^2) \tag{25}$$

$$= \mathbf{W}^T \left(\mathbf{W}^T \boldsymbol{\delta}^2 \cdot g'(\mathbf{a}^3)\right) \cdot g'(\mathbf{a}^2) \tag{26}$$

$$\vdots \tag{27}$$

$$= \mathbf{W}^T \left(\mathbf{W}^T \left(\left(\ldots (\mathbf{W}^{out})^T \boldsymbol{\delta}^L \ldots\right) \cdot g'(\mathbf{a}^3)\right) \cdot g'(\mathbf{a}^2)\right) \tag{28}$$

We can use the fact that $g' = \tanh' < 1$ to find an upper bound on the Euclidean norm of this error

$$\|\boldsymbol{\delta}^1\| \leq \|\mathbf{W}\|^{L-1} \|(\mathbf{W}^{out})\| \|\boldsymbol{\delta}^L\|.$$

We can rearrange this inequality to make explicit the role of $\bar{\sigma}(\mathbf{W})$, where the largest singular value of $\mathbf{W}$ is:

$$\|\boldsymbol{\delta}^1\| \sim \bar{\sigma}(\mathbf{W})^L \|\boldsymbol{\delta}^L\|, \tag{29}$$

and use the sufficient condition for the echo state property described as: $\bar{\sigma}(\mathbf{W}) < 1$. Hence, for an Echo State Network $\|\boldsymbol{\delta}^1\| = \bar{\sigma}(\mathbf{W})^L \|\boldsymbol{\delta}^L\| \to 0$ for $L \to +\infty$.

In summary, after the error has been back-propagated through the recurrent layers $n$ times, the magnitude of the gradient error used to update the word feature $f(\mathbf{u}_n)$ conditioned on the prediction of $\mathbf{u}_{t+n}$ tends to decrease geometrically. It is thus possible to approximate the gradient by taking into account only a limited number of time-steps. In fact we chose to compute the gradient only taking into account one pass through the recurrent layer. This allows reducing the computational burden of the feature update and makes the architecture of the proposed network close to a simple feed forward network with memory.

3.3. **Multiple linear readouts.** The second improvement that we propose is to adapt the form of the readout to the Markovian organisation of the reservoir. Since activations tend to cluster in regions corresponding to $n$-grams, using a readout capable to adapt its resolution to increasingly precise regions which is expected to greatly improve the accuracy of the model.

3.3.1. *Mixture of experts.* The mixture of experts model [3, 7] is a well known model in which the input data is first partitioned into soft clusters, each corresponding to an "expert domain", before being processed by specialized sub-models. Each expert can be a linear regression, a multinomial logit or any simple model for classification or regression. Model definition. In the mixture of experts model the conditional probability $P(\mathbf{y}|\mathbf{x})$ of generating $\mathbf{y}$ is expressed as a mixture of expert densities $P(\mathbf{y}|\mathbf{x}, z)$. The mixing coefficients $P(z|\mathbf{x})$ are multinomial probabilities depending on the input data $\mathbf{x}$. The total probability has the form:

$$P(\mathbf{y}|\mathbf{x}) = \sum_z P(z|\mathbf{x}) P(\mathbf{y}|\mathbf{x}, z) \tag{30}$$

$$= \sum_z P(\mathbf{y}, z|\mathbf{x}). \tag{31}$$

The marginalization over the latent variable $z$ is used to construct a complex (multimodal) distribution $P(\mathbf{y}|\mathbf{x})$ by combining simpler (unimodal) expert distributions $P(\mathbf{y}|\mathbf{x}, z)$.

The term $P(z|\mathbf{x})$ is called a gating model. Its task is to decide which expert is going to be applied to the prediction of $\mathbf{y}$ according to $\mathbf{x}$. The gating part of the model constructs a soft partition (see Figure 2) of the input space and assigns different experts $P(\mathbf{y}|\mathbf{x}, z)$ to different regions. For example, when words are input to the Echo State Network, the reservoir constructs a representation of the history of words in its activation space. The role of the gates in that case will be to cluster the reservoir activation space and assign a different expert predictor to each region. This clustering is soft in the sense that each region is assigned every expert with different probabilities. Figure 3 compares the architecture of an ESN with a mixture of experts readout to a conventional ESN.

The multinomial mixing coefficients of the gate model and the experts distributions are parameterized by a multinomial logit function:

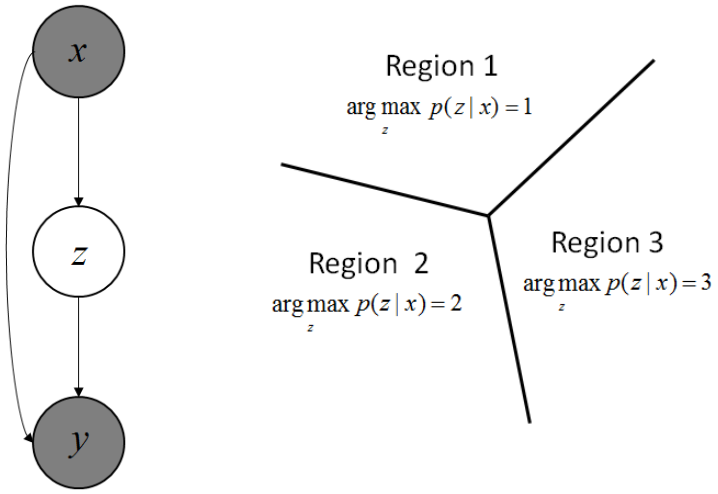
$$P(z = i|\mathbf{x}) = \frac{\exp(\boldsymbol{\Theta}_{\mathbf{g,i}}^{\mathbf{T}} \mathbf{x})}{\sum_k \exp(\boldsymbol{\Theta}_{\mathbf{g,k}}^{\mathbf{T}} \mathbf{x})}, \tag{32}$$

where $\boldsymbol{\Theta}_{g,i}$ are parameter vectors and

$$P(\mathbf{y}|\mathbf{x}, z = i) = \frac{\exp(\boldsymbol{\Theta}_{\mathbf{e,i}}^{\mathbf{T}} \mathbf{x})}{\sum_k \exp(\boldsymbol{\Theta}_{\mathbf{e,k}}^{\mathbf{T}} \mathbf{x})}, \tag{33}$$

where $\Theta_{e,i}$ is a vector. Both the gating and expert distributions could be chosen to be gaussian: the sub-model would then simply be linear regressions.

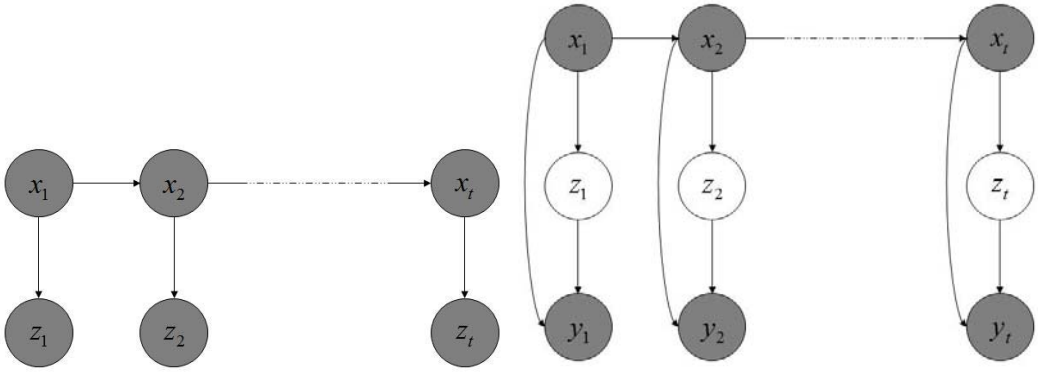
Since, the activation of the reservoir $\mathbf{x}$ is a representation of the past history of a sentence, it is expected that the gating model will be able to extract interesting linguistic features. Since the features introduced in the previous section are intended to re-organize the activation of the recurrent layer, the gating model is expected to give a higher level information on the kind of representation developed as discussed in 4.3. Those features will be extracted automatically during the training phase to maximize the likelihood of the data. The mixture of experts model and its extension the hierarchical micture of experts are thought to be good readout candidates since the activation of the reservoir is




(a) Mixture of experts model

(b) Experts' regions

FIGURE 2. The mixture of experts model and the way it clusters the input space into regions assigned to experts




(a) A conventional echo state network unrolled in time

(b) The mixture of experts model applied to an echo state network

FIGURE 3. The input $\mathbf{u}_t$ is omitted for clarity. (a) At each time-step the reservoir activation $\mathbf{x}_t$ is read directly by a linear regression. (b) At each time-step the reservoir activation $\mathbf{x}_t$ is read by a different expert according to the value of the latent variable $z_t$.

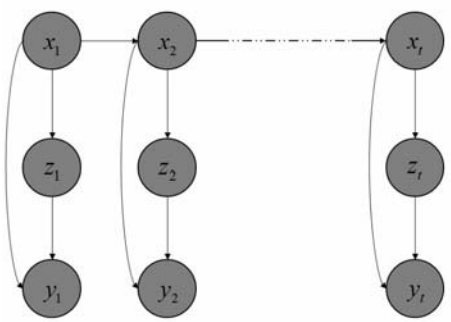



FIGURE 4. Supervised mixture of experts

fractal [14]. The gating nodes allow the model to have a better precision by focusing on a specific region of the activation space. By stacking many layers as in the hierarchical mixture of experts the fractal precision of the reservoir can thus, in principle, be attained. Learning with Expectation-Maximization. The parameters $\Theta$ of the model are learned by maximizing the log-likelihood of the data. The samples $(\mathbf{x}_n, \mathbf{y}_n) \in \mathcal{X}$ are assumed to be independent[1]:

$$\ell(\mathcal{X}, \Theta) = \ln \sum_{z=1}^{K} p(z|\mathbf{x}_n) p(\mathbf{y}_n, z|\mathbf{x}_n). \tag{34}$$

Although the gating and experts models[2] $p(z|\mathbf{x})$ and $p(\mathbf{y}, z|\mathbf{x})$ can be learned easily when they are used as standalone models (by learning a simple linear regression model or a logit regression), the summation inside the logarithm that appears in the Mixture of Experts model renders the learning of the parameters more complex. However, using the Maximization-Expectation algorithm allows to express a lower bound on the log-likelihood in which the different sub-model can be estimated independently as described in [9].

3.3.2. *Supervised mixture of experts.* The mixture of experts model can be simplified by specifying which expert should predict the next word in a supervised way. Instead of letting the model cluster be the input space according to features it discovers itself, we can specify the expert that the model must rely on for the prediction of the next word by assigning a value to the $z^t$ for every $t$. This actually simplifies the learning properties of the algorithm: the training is now totally supervised in the sense that instead of a partially observed dataset $\mathcal{X} = \{\mathbf{X}, \mathbf{Y}\}$ the algorithm can now have access to the fully observed dataset $\mathcal{X} = \{\mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$.

However, we have to choose a way to automatically specify the expert. This amounts to restricting the kind of feature that the algorithm is paying attention to by actually choosing the features. For example, we could choose to assign a different expert according to the position of the current word in the sentence. Thus, one expert could be a specialist at predicting the beginning of a sentence while another would be responsible for sentences endings. Many other linguistic characteristics could be used to choose between experts; this allows adding hand crafted features for selecting experts.

Another way of selecting the experts is to rely on the word to be predicted. For example, using the representation developed in the feature layer, we can cluster words and assign an expert to each cluster of words. In fact, this trick is used in [13] to speed up the training

[1]This is not the case for Echo State Networks but we make this simplification nevertheless in order to decouple the samples and make the training easier.

[2]In order to keep the notations uncluttered we often omit to specify explicitly the dependency of the different probability distribution on their parameters.

of a neural language model: each expert is assigned to a subset of the vocabulary. The probability of the next word is computed according to

$$P(w^{t+1}|w^1 \dots w^t) = \sum_{w \in V} p(z_w|w^1 \dots w^t) p(w|z_w, w^1 \dots w^t) \tag{35}$$

with the supplementary condition that $p(w|z_{w'}) = 0$ if $w \neq w'$. Each expert has to focus on a subset of the vocabulary. To minimize the dimensionality of the system and reduce the training time, each one of the $\sqrt{|V|}$ experts may focus on $\sqrt{|V|}$ words. Thanks to this trick, when the size of the vocabulary grows like $|V|$, it is possible to train the gating and experts models on vector of size only $\sqrt{|V|}$. The gain in training time is very sensible if the vocabulary is large and this trick allows to train a network efficiently on a huge number of words in a reasonable time [11, 12].

Finally, it is also possible to keep the latent variable in the mixture of experts model and add another level of nodes each focusing on a subset of the vocabulary. In fact the architecture can be stacked by any number of time as in the hierarchical mixture of experts model [9]. This has not been investigated in our experiments.

## 4. **Results.**

This section discusses the performance of the different models against the baseline model described in [18].

### 4.1. **Settings.**

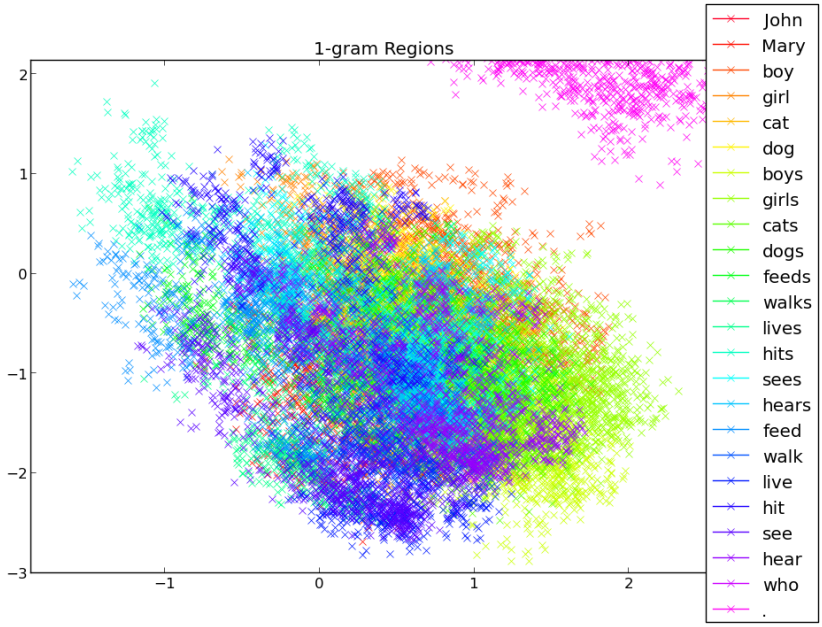
The models were trained on an artificial corpus generated using the Elman grammar. The training set comprised of 5500 sentences of length 3 to 9 averaging 6.7 words. The perplexity of the model on another set of 5500 sentences was used to assess the performance increase. Some sentences were common to the training and testing corpora because of the random construction method although the size of the grammar productions up to 9 words per sentence ensured that many sentences were unique in each corpus. In all the experiment, the whole dataset was processed as a long sequence of words. The state of the reservoir was not reset after each sentence. For this reason, we use perplexity rather than cosine similarity to measure the performance of the different models.

Several trials were realized by changing the size of the recurrent layer while the size of the features was kept constant and equal to two. The spectral radius was set to 0.97 in all experiments for this value gave good results in preliminary testing. The input scaling of the reservoir was 1. The model was trained to learn a feature representation of the input on ten steps of gradient descent with a learning rate of 0.01 for the output weights and 0.05 for the features. After that the features were kept fixed and the best linear regression was learned in a one-shot way. In the logistic and mixture of expert case, the features were also learned first and the Newton-Raphson maximization was then performed to learn the readout function.

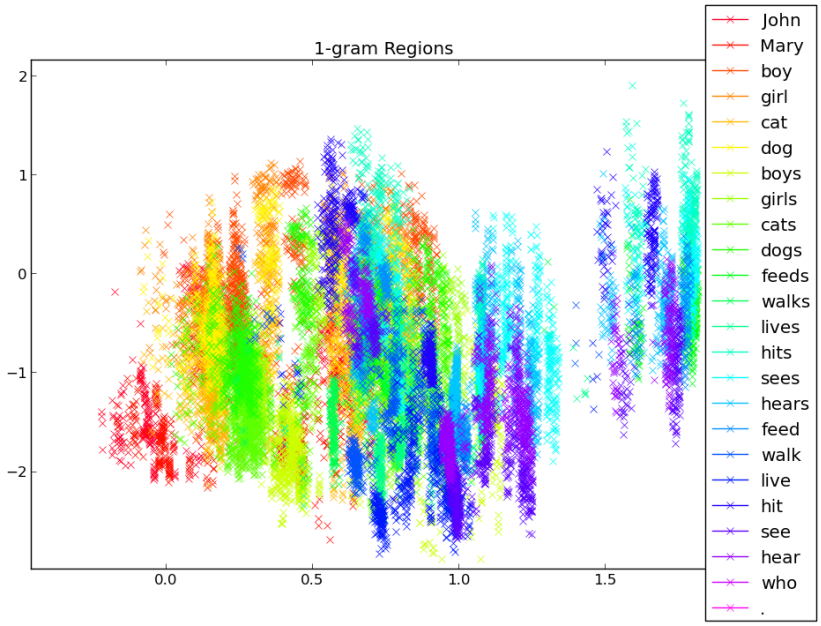
### 4.2. **Reservoir reorganization.**

The introduction of word features before the recurrent layer in turn leads to a novel organization of the reservoir activation pattern. In Figure 5 the activation pattern of the same reservoir driven by one-hot word representation and by distributed features can be compared. Since the real pattern lies in a 100 dimensional space, we used principal component analysis to project it on a 2 dimensional space to be able to visualize it.

Comparing the two activation patterns it seems that the word features allow separating more sharply the activations corresponding to different words. The reservoir should thus be easier to read. Also, the syntactic organization of the feature space described in [14] (nouns and verbs tend to belong to different regions of the reservoir activation space) is present in the reservoir activations although it is less marked than in the word features space.




(a) Without features




(b) With features

FIGURE 5. Comparison between the activation of the reservoir driven by one-hot representations and the same reservoir driven by learned features. The activation of the words "who" and the sentence ending marker "." have been omitted in the figure for clarity.

4.3. **Mixture of experts features.** It was expected that the gating model would be able to make use of the sharper separation induced by the feature layer in the reservoir activation to perform a clever clustering. It is indeed the case that the gating model partitions the reservoir space and assigns different regions to different experts, however, the features discovered by the mixture of experts model seem to depend greatly on the quality of the reservoir. Although in some cases it seems that the features discovered by

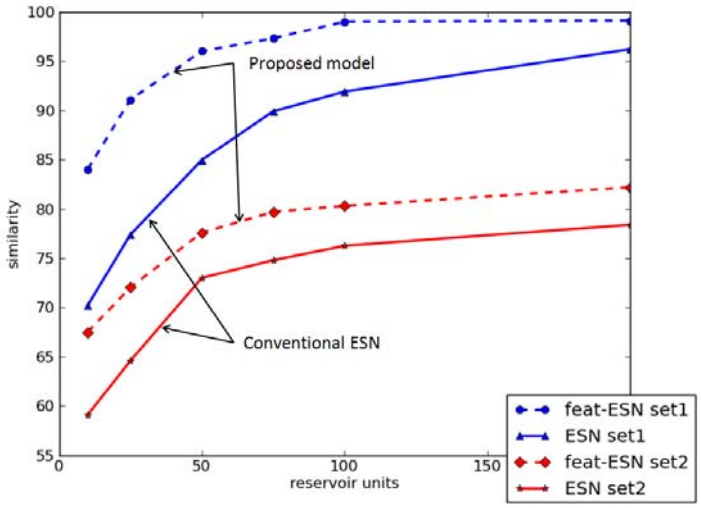



FIGURE 6. Comparison of the conventional echo state model and the model augmented with a pre-recurrent feature layer. The blue lines correspond to the first dataset while the red ones are the model tested on the second dataset. The performance of the model with features is shown with dashed lines.

the gates have a sensible interpretation in term of the linguistic structure of the sentences, in some other cases it was difficult to link the clusters to a good interpretation. In several instances of the experiment the clustering performed by the gating models seemed to take into account the length of the sentences. Short sentences tend to be assigned to an expert while longer sentences are assigned to another expert. Still the partition remains hard to interpret with certainty.

4.4. **Cosine similarity.** Figure 6 shows the improvement in accuracy induced by the introduction of a pre-recurrent feature layer. The baseline model corresponding to a traditional ESN is used in [18]. The cosine similarity of the networks prediction with the real next word probability is used to measure the performance and rescaled from 0 to 100. As can be expected the introduction of a feature layer is especially interesting for networks with small reservoirs but remains significant even for large ones.

4.5. **Perplexity.** We now compare the accuracy of different models using perplexity. The accuracy of an Echo State Network augmented with a pre-recurrent feature layer is shown in Figure 7. Four different readout are compared: linear, multivariate logistic, a mixture of experts model with 10 experts and finally a supervised mixture of 10 experts with each gating node assigned to a subset of the vocabulary.

In terms of accuracy, it can be seen that the traditional linear readout performs worst. Simply using a logistic regression as a readout already gives a good improvement. This may be due to the fact that the output of the linear readout is not a real probability distribution and must be renormalized appropriately. The mixture of experts model performs extremely well and is close to the best perplexity even with a reservoir of only 25 units. The supervised mixture of experts model on the other hand performs slightly worse than a mere logistic regression. In the case of the mixture of experts and supervised mixture of experts, the convergence of the training algorithms are more difficult to attain. The mixture of expert model appears to be especially sensitive to the type of optimization that is used and could achieve satisfactory perplexity only with a second order

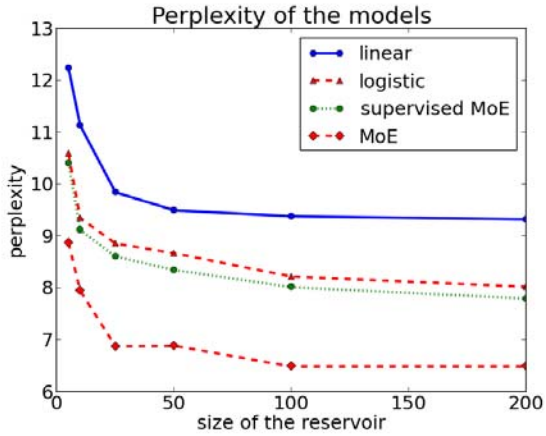



FIGURE 7. Perplexity of the different models on the second training set processed as a single sequences of words

optimization technique. Moreover, for small reservoirs, the inversion of the hessian used in the Newton-Raphson method was relying on a truncated conjugate gradient whereas for larger reservoir the conjugate gradient had to be run for many steps.

5. **Conclusion.** The echo state network model augmented with a feature layer and multiple readouts can serve as a basis for a variety of more elaborate language models. It can be seen as a hybrid between a simple recurrent neural network (the part constructing the sequence representation) and a feed-forward neural network (the part reading the reservoir). These models are tested on simple datasets and show significant improvement over the conventional architecture that was used in [18].

The particularities of the proposed models and their practical advantages are:

- When conventional echo state networks are augmented with a pre-recurrent feature layer the activation pattern in the reservoir becomes sharper than in the absence of a pre-recurrent processing of the input and the readout can extract information more easily. This greatly improves the accuracy of the language model.
- In some cases the clustering of the reservoir activation patterns done by the gating model of mixture of experts can be given an intuitive interpretation and the hierarchical architecture of the system in turns leads to a hierarchy of features. In other cases however the features discovered are not easily interpretable.
- The features are learned using a truncated back-propagation through time algorithm that can be justified based on the echo state property of the reservoir. This truncation greatly speeds up the training time making the model potentially usable on large datasets. This palliates potential limitation of traditional ESNs that require a large number of units and cannot be trained on large datasets due to the excessive memory footprint.
- The gating and experts model can be learned independently using the Expectation-Maximization algorithm. This can be advantageous when training the algorithm in a distributed environment or to make the best use of machines with several cores though it was not tested for this study.

The great advantage of conventional echo state networks remains that it is possible to train them in a very fast and efficient way using one shot learning when the training dataset fits in memory. On the contrary, the parameters of the models we proposed are estimated using iterative method thus requiring more effort to achieve convergence. In particular, Newton-Raphson's method becomes quickly impractical when the number of

output dimensions increases. One solution to this dimensionality problem is to apply the trick proposed in [13] that can be viewed as a special case of "supervised mixture of experts". In that case it is also possible to consider parameterizing every experts and gating models to be linear regressions thus allowing each subsystem to be trained in one shot.

The algorithms proposed have been tested on relatively small artificial datasets and their performance on real data remains uncertain. Applying the augmented echo state networks models to large datasets and comparing their performances to other neural network language models is the next step in the study of their performance.

## REFERENCES


[1] Y. Bengio, R. Ducharme, P. Vincent and C. Janvin, A neural probabilistic language model, *J. Mach. Learn. Res.*, vol.3, pp.1137-1155, 2003.

[2] Y. Bengio, P. Simard and P. Frasconi, Learning long-term dependencies with gradient descent is difficult, *IEEE Transactions on Neural Networks*, vol.5, no.2, pp.157-166, 1994.

[3] C. M. Bishop and M. Svens, Bayesian hierarchical mixtures of experts, *Proc. of the 19th Conference on Uncertainty in Artificial Intelligence UAI 2003*, vol.108, no.2, pp.1-8, 2003.

[4] S. F. Chen and J. Goodman, An empirical study of smoothing techniques for language modeling, *Proc. of the 34th Annual Meeting on Association for Computational Linguistics*, pp.310-318, Stroudsburg, PA, USA, 1996.

[5] R. Collobert and J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, *Proc. of International Conference on Machine Learning*, 2008.

[6] J. L. Elman, Distributed representations, simple recurrent networks, and grammatical structure, *Mach. Learn.*, vol.7, pp.195-225, 1991.

[7] R. A. Jacobs, M. Jordan, S. Nowlan and G. Hinton, Adaptative mixtures of local experts, *NBeural Computation*, vol.3, no.1, pp.79-87, 1991.

[8] H. Jaeger, The echo state approach to analysing and training recurrent neural networks – With an erratum note, *Tecnical Peport GMD Report 148*, 2001.

[9] M. I. Jordan and R. A. Jacobs, Hierarchical mixtures of experts and the EM algorithm, *Neural Computation*, vol.6, no.2, pp.181-214, 1994.

[10] S. Katz, Estimation of probabilities from sparse data for the language model component of a speech recognizer, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol.35, no.3, pp.400-401, 1987.

[11] T. Mikolov, A. Deoras, D. Povey, L. Burget and J. Cernocky, Strategies for training large scale neural network language models, *Proc. of ASRU*, 2011.

[12] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky and S. Khudanpur, Extensions of recurrent neural network language model, *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.5528-5531, 2011.

[13] F. Morin, Hierarchical probabilistic neural network language model, *Proc. of the International Workshop on Artificial Intelligence and Statistics*, pp.246-252, 2005.

[14] A. Rachez and M. Hagiwara, Augmented echo state networks with a feature layer and a nonlinear readout, *Proc. of International Joint Conference on Neural Networks*, pp.1-8, 2012.

[15] R. Socher, B. Huval, C. D. Manning and A. Y. Ng, Semantic compositionality through recursive matrix-vector spaces, *EMNLP (Mv)*, 2011.

[16] R. Socher, C. D. Manning and A. Y. Ng, Learning continuous phrase representations and syntactic parsing with recursive neural networks, *Science*, vol.321, pp.1-9, 2010.

[17] P. Tino and B. Hammer, Markovian architectural bias of recurrent neural networks, *IEEE Transactions on Neural Networks*, vol.15, pp.6-15, 2002.

[18] M. H. Tong, A. D. Bickett, E. M. Christiansen and G. W. Cottrell, 2007 special issue: Learning grammatical structure with echo state networks, *Neural Netw.*, vol.20, pp.424-432, 2007.

[19] *Reservoir Computing Homepage*, http://reservoir-computing.org.