

Utilizando *Reservoir Computing* para a Previsão da Potência do Vento

Bruna Galle
Escola Politécnica de Pernambuco
Universidade de Pernambuco
Brasil
Email: bcga@ecomp.poli.br

Dr. Mêuser Jorge Silva Valença
Escola Politécnica de Pernambuco
Universidade de Pernambuco
Brasil
Email: meuser@ecomp.poli.br

Abstract—Um dos principais desafios da atualidade é a crescente demanda de energia mundial e, a fim de suprir essa necessidade, as fontes de energia mais utilizadas são o petróleo, gás natural e carvão mineral. O grande problema com essas fontes deve-se ao fato de, além de serem extremamente poluidoras, elas são de origem não renovável. Por causa disso, energias renováveis estão se tornando cada vez mais essenciais para a humanidade e entre elas, está o vento e sua escolha é uma das mais promissoras. Os parques eólicos tem seu potencial diretamente ligado à potência do vento, sendo necessária boas estimativas dessa variável para a construção de estratégias e planejamentos eficientes. Entretanto, essa tarefa apresenta grandes dificuldades devido às complexas características do vento, como a alta variabilidade de sua velocidade e direção. Este trabalho tem como objetivo utilizar a técnica de *Reservoir Computing* para a previsão da potência do vento e, pelo fato da *Multi-Layer Perceptron* ser a mais utilizada para tal fim, realizar a comparação entre esses dois tipos de Redes Neurais Artificiais e analisar qual possui o melhor desempenho.

I. INTRODUÇÃO

Um dos principais desafios da atualidade é a crescente demanda de energia mundial e, a fim de suprir essa necessidade, as fontes de energia mais utilizadas são o petróleo, gás natural e carvão mineral. O grande problema com essas fontes deve-se ao fato de, além de serem extremamente poluidoras, elas são de origem não renovável, ou seja, irão exaurir-se da natureza dentro de alguns anos [1].

Segundo a Agência Internacional de Energia (AIE), caso não se reduza a média de consumo registrada nas últimas décadas, as reservas mundiais de petróleo e gás natural deverão se esgotar em 100 anos, e de carvão, em 200 [1]. Dessa forma, tornou-se essencial a utilização de fontes renováveis e que ajudem no combate à degradação sofrida pelo meio ambiente.

As energias renováveis, pelos motivos citados acima, estão se tornando cada vez mais essenciais para a humanidade e sua maior vantagem é por serem seguras, limpas, abundantes e conseqüentemente não impactarem de forma negativa nas questões ambientais. Entre as diversas fontes disponíveis no mundo está o vento e sua escolha é uma das mais promissoras. Isso se explica devido a sua disponibilidade constante em qualquer lugar e por sua produção ser considerada hoje de custo competitivo [2].

No entanto, devido à aleatoriedade da geração eólica, não é possível garantir um montante fixo de energia ao sistema elétrico. Adicionado a isso está o investimento cada vez mais

alto de diversos governos neste tipo de energia para suprir o alto consumo de eletricidade nos últimos anos. Desta forma, e a fim de ajudar os países cuja matriz energética passou a incluir o vento como fonte alternativa, a previsão de energia eólica mostra-se fundamental para definição de estratégias e planejamentos adequados, eficientes e não dispendiosos. Essa previsão depende principalmente da potência do vento. Existem diversos modelos utilizados para realizá-la, vários deles incluindo inteligência artificial. Este trabalho tem como objetivo utilizar uma arquitetura de Rede Neural Artificial (RNA) chamada *Reservoir Computing* (RC) e analisar seu desempenho nesta tarefa.

Embora já existam modelos utilizando tipos mais comuns de RNAs, como o *Multi-Layer Perceptron* (MLP), o *Reservoir Computing* foi escolhido por possuir uma arquitetura em que os neurônios artificiais estão interligados e organizados de forma mais semelhante ao cérebro humano (metáfora que é a origem das RNAs, como o próprio nome indica).

Essa característica da arquitetura do RC permite que essa técnica represente sistemas com comportamento dinâmico, o que redes neurais estáticas como o MLP tem dificuldade ao fazê-lo [3]. Conseqüentemente, aprender as características temporais dinâmicas que representam a série histórica da potência do vento torna-se uma tarefa mais adequada para o RC.

Devido a isso, espera-se que seus resultados de desempenho das previsões sejam melhores dos que aqueles obtidos pelas outras RNAs. Logo, a previsão seria mais precisa e aumentaria a eficiência no planejamento e uso da energia eólica, incentivando seu uso em diversos lugares e preservando o meio ambiente.

II. *Reservoir Computing*

Além das arquiteturas com alimentação adiante (*feed-forward*), como o MLP, utilizadas amplamente na previsão de séries temporais, começaram a surgir as Redes Neurais Recorrentes (RNR). Neste novo tipo de rede, existe a adição de conexões recorrentes às já existentes arquiteturas *feedforward*.

Essas conexões transformam o sistema em um sistema dinâmico complexo e ainda mais adequado para a resolução de problemas temporais. No caso deste projeto, torna-se uma opção atraente visto que o problema a ser resolvido, a previsão da potência do vento, é de natureza temporal. A Figura II

apresenta a estrutura de uma rede alimentada adiante e de uma rede recorrente [4].

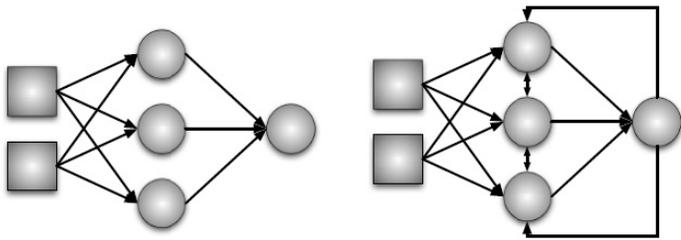


Fig. 1. Estrutura de uma rede alimentada adiante (esquerda) e de uma rede recorrente (direita)

[Fonte: [3]]

As RNRs são modelos computacionais capazes de criar internamente a memória necessária para armazenar o histórico dos padrões de entrada através das suas conexões recorrentes [4].

Em 2001, uma nova proposta para o design e treinamento de RNRs foi sugerida, de forma independente, por Wolfgang Maass com o nome *Liquid State Machine* (LSM) e por Herbert Jaeger com o nome *Echo State Networks* (ESN). Verstraeten propôs a unificação dessas duas abordagens em um único termo chamado *Reservoir Computing*. Desde então, RC passou a ser adotado na literatura como um nome genérico para sistemas de aprendizado que consistem de uma rede recorrente dinâmica com simples nós computacionais combinada com uma simples função de saída [5].

Um sistema de *Reservoir Computing* é composto por duas partes principais: o *reservoir* e uma camada de saída linear. O *reservoir* é um sistema dinâmico não linear com uma topologia recorrente de nós de processamento. As conexões entre os nós são geradas aleatoriamente e são globalmente reescaladas a fim de se alcançar um estado dinâmico adequado. Uma propriedade importante do RC é que o *reservoir* possui pesos fixos, isto é, não se faz necessário o seu treinamento. Apenas a camada de saída é treinada e é por este motivo que ela possui uma função de saída. Essa função pode ser, por exemplo, um classificador linear ou um algoritmo de regressão [5].

O fato de apenas a camada de saída precisar ser treinada possibilita, mantendo-se as mesmas entradas, o uso do mesmo *reservoir* para a solução de diferentes tarefas simultaneamente.

Uma característica interessante do RC é aquela baseada nas ESNs, a chamada propriedade do eco. Essa propriedade define que os efeitos de um estado anterior $x(n)$ e um valor de entrada em um futuro estado $x(n+1)$ devem desaparecer gradualmente com o passar do tempo k (isto é, $k \rightarrow \infty$) e não devem persistir, nem serem amplificados.

Devido às conexões recorrentes, informações sobre entradas passadas são armazenadas na rede. Por causa disso, a rede contém um rico conjunto de transformações não lineares e misturas de sinais de entrada de tempos passados e do presente (os chamados ecos).

A. Criação e Uso da Técnica de RC

Na explicação a seguir, admite-se que o sistema RC é constituído de N nós do *reservoir*, M entradas e P saídas.

1) Criação das entradas e das conexões do *reservoir*:

- 1) Cria-se uma matriz de peso W_{in} $M \times N$ da camada de entrada para o *reservoir*. Os pesos são gerados a partir de uma distribuição aleatória ou de um conjunto discreto. Se todos os sinais de entrada devem alimentar o *reservoir*, então toda a matriz é preenchida por valores diferentes de zero. Caso contrário, haverá elementos nulos.
- 2) Cria-se uma matriz de peso W_{res} $N \times N$ com as interconexões do *reservoir*. Os valores dos pesos são novamente gerados a partir de uma distribuição (exemplo, a gaussiana) ou um conjunto de valores discretos (exemplo, $-1, 1$).
- 3) Reescala-se a matriz de pesos W_{res} globalmente a fim de o *reservoir* possuir uma estabilidade numérica. O modo mais comum de fazer isso é ajustando o raio espectral de W_{res} . O raio espectral de uma matriz é o maior valor absoluto entre os autovalores. Um valor perto de 1 é geralmente considerado um bom ponto inicial para a otimização de ESNs.

2) Simulando o *reservoir* e treinando e testando as saídas:

- 1) Criar uma base de dados BD e dividi-la em três subconjuntos: treinamento, validação cruzada e testes.
- 2) O estado da rede no tempo k é denotado como $x[k]$ e uma entrada no mesmo tempo como $u[k]$. Para cada amostra, inicializamos $x[0] = 0$. Antes de iniciar-se o treinamento, os primeiros n ciclos são chamados de “*warm up*”, onde a rede neural esquece seus estados iniciais e perde a influência do valor zero (0) atribuído aos estados iniciais. Ao fim do “*warm up*”, acontece o treinamento e a rede é simulada recursivamente.
- 3) Depois que todas as amostras forem simuladas, as matrizes de estado do conjunto de treinamento são concatenadas em uma matriz maior A .
- 4) Calculam-se os pesos da camada de saída. Neste projeto, para realizar este cálculo, foi utilizada a matriz inversa de Moore -Penrose ou pseudoinversa, pois a matriz A não é quadrada. Esses cálculos serão feitos pela rotina do Java chamada JAMA de forma automática.
- 5) Após um exemplo de entrada ser treinado, é feita a validação cruzada a fim de checar se o treinamento já pode ser finalizado. O Erro Médio Quadrático (EMQ) é calculado e armazenado a cada validação cruzada feita.
- 6) Ao fim do treinamento, simula-se a rede com o conjunto de teste e calcula-se o Erro Percentual Médio Absoluto (EPMA). Esses erros são armazenados para posterior realização de testes estatísticos. Neste projeto, foi utilizado o Erro Médio Absoluto Normalizado (NEMA).

Através dos NEMAs calculados para o desempenho da rede com a técnica RC e a MLP, será possível a comparação de qual arquitetura é a melhor escolha para a previsão da potência do vento.

III. METODOLOGIA

A. Base de Dados

A base de dados utilizada nos experimentos foi cedida pelo Operador Nacional de Sistema Elétrico (ONS). O ONS é o órgão responsável pela coordenação e controle da operação das instalações de geração e transmissão de energia elétrica no Sistema Interligado Nacional (SIN), sob a fiscalização e regulação da ANEEL [6].

Os dados de velocidade média dos ventos são diários e o período em que foram medidos e coletados vai de 01 de dezembro de 2011 até 31 de julho de 2012. Eles foram observados de 30 em 30 minutos e os dados disponíveis são: a velocidade média do vento, a direção e a potência. Dentre eles, será utilizado apenas a potência do vento, pois notou-se que apenas essa variável alcança bons resultados.

Cada parque eólico possui uma capacidade de potência instalada associada a ele. Esse valor indica o máximo de potência que aquele parque pode produzir. No caso do parque ao qual essa base de dados pertence, a capacidade é de 54.61 MW.

B. Pré-processamento dos dados

O primeiro passo durante a fase de pré-processamento é a normalização dos valores. Essa etapa tem como objetivo evitar que valores altos influenciam de forma demasiada nos cálculos da RNA enquanto os valores baixos passam despercebidos. Ou seja, é feita para garantir que as variáveis em intervalos diferentes recebam a mesma atenção durante o treinamento. Além disso, as variáveis devem ter seus valores proporcionais aos limites da função de ativação usada na camada de saída. Se a função de ativação escolhida for a sigmoide logística, seus valores estão limitados entre [0 e 1], logo os dados são geralmente normalizados entre [0,10 e 0,90] ou [0,15 e 0,85] [7].

A normalização é feita através da fórmula descrita na equação 1:

$$y = \frac{(b - a)(x_i - x_{min})}{(x_{max} - x_{min})} + a \quad (1)$$

Onde:

- y = valor normalizado;
- x_i = valor original;
- x_{min} = valor mínimo de x ;
- x_{max} = valor máximo de x ;
- a e b = limites escolhidos. Neste trabalho, $a = 0,15$ e $b = 0,85$.

C. Medida do Desempenho da Rede

Na previsão da potência do vento, as medidas de erros usuais, como o EPMA e o Erro Médio Absoluto (EMA), são dados como um percentual da capacidade instalada de determinado parque eólico. Neste trabalho, definiu-se que o desempenho da rede seria medido através do Erro Médio Absoluto Normalizado (NEMA).

D. Predizendo Velocidade do Vento com MLP

Embora seja bastante utilizada em diversas pesquisas, o MLP necessita que vários de seus parâmetros sejam configuráveis e a escolha de cada um influencia diretamente no resultado final da previsão.

Abaixo, seguem os principais parâmetros do MLP e do algoritmo *Backpropagation*:

- Número de neurônios na camada de entrada;
- Número de neurônios na camada escondida (apenas uma camada escondida);
- Número de neurônios na camada de saída;
- Função de ativação;
- Critério de Parada;
- Taxa de Aprendizado;
- Momento.

A quantidade de entradas foi definida após a análise do impacto de uma quantidade maior, como 48, ou menor de entradas, como 7, e se faziam diferença na previsão da potência do vento. Notou-se que o resultado final não variava de forma significativa e a fim de acelerar o treinamento, escolheu-se a quantidade de 7 entradas.

Como deseja-se a previsão de 2 dias à frente, a saída é sempre 48 valores de potência de vento.

O algoritmo utilizado é o *Backpropagation* e a função de ativação escolhida para os neurônios é a sigmoide logística. Essa função retorna valores no intervalo [0, 1].

O critério de parada adotado foi o de validação cruzada, com 50% do conjunto de valores para treinamento, 25% para validação cruzada e os restantes 25% para teste.

Diversos testes foram realizados para definição da taxa de aprendizado, do momento e do número de neurônios na camada escondida, sendo que os melhores desempenhos corresponderam aos valores de 0,8 para a taxa de aprendizado, 0,2 para o momento e 6 neurônios na camada escondida.

O MLP utilizada foi implementado na linguagem de programação JAVA no ambiente de desenvolvimento Eclipse [8].

E. Predizendo a Velocidade do Vento com RC

Assim como o MLP, a técnica de *Reservoir Computing* possui diversos parâmetros que necessitam de configuração. Como é uma área de pesquisa recente, a escolha dessas configurações não pode ser considerada ideal e muitas vezes é realizada de forma aleatória. Uma forma é avaliar cada valor escolhido e

determinar se ele foi melhor ou pior para o desempenho da rede. Repete-se esse processo até ser encontrado um valor considerado ótimo, o que não necessariamente significa o melhor, algo que seria possível alcançar caso houvesse uma forma mais adequada de defini-lo.

Abaixo seguem os parâmetros cujas configurações foram necessárias serem feitas ao longo deste projeto:

- Quantidade de neurônios na camada de entrada;
- Quantidade de neurônios na camada de saída;
- Quantidade de neurônios do *reservoir*;
- Função de ativação do *reservoir*;
- Função de ativação da camada de saída;
- Inicialização dos pesos;
- Taxa de conexão do *reservoir*;
- Quantidade de ciclos do “*warm up*”;
- Critério de parada.

A quantidade de entrada permanece a mesma utilizada no MLP, visto que é preciso que esse parâmetro mantenha-se com o mesmo valor da rede neural anterior a fim de realizar-se os testes estatísticos. O mesmo aplica-se para a quantidade de saídas. Ou seja, o RC terá 48 saídas para prever 2 dias à frente.

A quantidade de neurônios no *reservoir* é um dos parâmetros para o qual não existe um critério fixo que o determine. Foi escolhido aleatoriamente e após a verificação do EPMA, ao fim de cada treinamento, o melhor número de neurônios. Observou-se que para a solução do problema em questão, o número ideal encontrado foi de 20 neurônios.

Como citado na seção II, os pesos da camada de entrada para o *reservoir* e os pesos do *reservoir* são gerados aleatoriamente a partir de uma distribuição aleatória.

Os estados do *reservoir* são inicializados como zero (0). Por causa disso, como citado na seção II, decidiu-se adicionar a rede uma fase denominada “*warm up*”. Durante o “*warm up*”, não há necessidade de encontrar os pesos da camada de saída, nem de calcular um valor de saída. Isso se explica pois esta fase “*warm up*” é apenas para atualizar os estados do *reservoir* e retirar a dependência do estado inicial. O número de ciclos do “*warm up*” escolhido foi de 10.

A taxa de conexão dos neurônios do *reservoir* foi de 50%. Ou seja, metade das ligações possuem pesos diferentes de zero (0) associadas a elas.

O critério de parada escolhido também foi o de validação cruzada, com 50% da base de dados para o conjunto de treinamento, 25% para a validação cruzada e os outros 25% para teste.

A função de ativação escolhida no *reservoir* foi a sigmoide logística e sua fórmula foi descrita na seção anterior. Na camada de saída, a função selecionada foi a linear.

Durante este projeto foi implementada uma Rede Neural com a técnica de RC na linguagem de programação JAVA e no ambiente de desenvolvimento Eclipse. O pseudocódigo 1 é a forma sintetizada de como funciona essa RNA.

Algoritmo 1: Pseudocódigo do RC

```
1 Definir a quantidade de neurônios na camada de entrada ;
2 Definir a quantidade de neurônios no reservoir ;
3 Definir a quantidade de neurônios na camada de saída ;
4 Gerar aleatoriamente os pesos da matriz Win entre -1 e 1;
5 Gerar aleatoriamente os pesos da matriz Wres entre -1 e 1;
6 Normalizar os pesos de Wres para que o raio espectral da matriz fique menor ou igual a 1;
7 while até o final da quantidade de ciclos de warm up
8   do
9     | atualiza os estados dos neurônios do RC;
10  end
11 while até que o critério de parada seja atingido do
12   | for cada valor do conjunto de entrada do
13     | atualiza os estados dos neurônios do RC;
14   end
15   Calcula a pseudo-inversa de Moore-Penrose para encontrar a matriz de pesos que conecta o RC à camada de saída;
16   for cada valor do conjunto de validação cruzada do
17     | atualiza os estados dos neurônios do RC;
18   end
19   Calcula os valores de saída do RC;
20   Calcula o EMQ;
21   Verifica se o critério de parada já foi atingido;
22 end
23 for cada valor do conjunto de testes do
24   | atualiza os estados dos neurônios do RC;
25 end
26 Calcula os valores de saída do RC;
27 Calcula o NEMA // para o caso de previsão da potência do vento;
```

F. Testes Estatísticos

Após os 30 treinamentos [9] com cada tipo de rede neural, foram realizados testes estatísticos com o objetivo de avaliar qual técnica apresenta o melhor desempenho na previsão da potência do vento ou se seus resultados podem ser considerados estatisticamente iguais.

Entre os diversos testes existentes na literatura, se encontram o teste T-student e o teste de Wilcoxon. Para aplicação do teste T-student deve-se começar utilizando o teste de Shapiro-Wilk com o objetivo de checar se os dados estão normalmente distribuídos e, caso sejam consideradas amostras normais, é realizado em seguida o teste F para verificar se a variância pode ser considerada como originada de uma mesma população. Ao passar nestes testes é aplicado o teste T-student para dados não emparelhados. Caso as amostras não sejam consideradas normais ou não passem no teste F, é aplicado o teste da Soma dos Postos de Wilcoxon.

IV. RESULTADOS

O treinamento da MLP e do RC foi realizado com várias configurações de entrada, como citado na seção anterior, até

o melhor resultado ser alcançado. Foram testadas 7, 24 e 48 entradas e como pode ser observado na tabela I, o NEMA não variou significativamente ao alterar-se a quantidade de entradas.

TABLE I. EXECUÇÕES PARA DETERMINAÇÃO DA QUANTIDADE DE ENTRADAS

[Fonte: elaboração própria]

Qtd. de Neurônios na Entrada	RC	MLP
7	18,04%	23,54%
24	18,47%	24,03%
48	18,23%	24,07%

Foram realizadas 30 simulações para cada topologia e a média dos NEMAs pode ser encontrada na tabela II.

TABLE II. A MÉDIA DOS NEMAS PARA AS DUAS ARQUITETURAS DE RNAs

[Fonte: elaboração própria]

Topologia	Média do NEMA
RC	18,02 %
MLP	24,47%

Em seguida, para cada conjunto de 30 simulações, foram realizados os testes estatísticos. Iniciou-se pelo teste de Shapiro-Wilk e utilizando o software R, que como padrão define o nível de significância como 0,05, observou-se que a amostra pertencente a MLP não provém de uma população normal, ou seja, não é normalmente distribuídas, uma vez que o p-value calculado durante o teste foi 0,005 para o MLP, ou seja, menor que o nível de significância.

A partir deste resultado, tornou-se desnecessário a realização do teste T de Student, cuja pré-condição para seu uso é de que as amostras fossem normais. Por causa disso, o próximo passo foi à aplicação do teste da Soma dos Postos de Wilcoxon.

Esse teste é não-paramétrico, ou seja, supõe-se que a distribuição dos dados não seja normal. O resultado deste teste mostrou que o p-value possui um valor muito menor do que o nível de significância, logo, refuta-se a hipótese nula, aquela que afirma que o desempenho das redes neurais com as arquiteturas RC e MLP são consideradas estatisticamente iguais, e conclui-se que o uso da técnica proposta de RC possui um melhor desempenho.

V. CONCLUSÃO

Este trabalho teve como principal objetivo prever a potência do vento para 2 dias a frente utilizando a técnica de *Reservoir Computing* e comparar com os resultados existentes providos por outros modelos de previsão. Como a rede neural MLP é a mais utilizada neste tipo de aplicação, essa foi a técnica escolhida e com a qual os resultados do RC foram comparados.

A fim de alcançar esse objetivo, foi implementada uma rede neural com a topologia RC e uma base de dados disponibilizada pelo Operador Nacional do Sistema Elétrico foi utilizada. Realizou-se diversas simulações com as duas arquiteturas e seus resultados foram comparados.

Através dos testes estatísticos, foi comprovado que o desempenho com o RC é melhor do que aquele encontrado pelo MLP e abre-se um campo de pesquisa para o estudo de redes neurais dinâmicas como o RC para previsão de séries temporais.

Como trabalhos futuros, pretende-se a realização de um estudo mais aprofundado dos parâmetros do RC a fim de encontrarem-se as melhores maneiras de defini-los e que possam influenciar de forma positiva no desempenho da RNA.

Por fim, é necessário obter-se bases de dados de outros lugares do país e do mundo e observar se a técnica de RC continua sendo a melhor escolha na previsão da potência do vento.

REFERENCES

- [1] W. de Cerqueira e Francisco. (2008) Fontes de energia. Acessado em 29 de agosto de 2012. [Online]. Available: <http://www.mundoeducacao.com.br/geografia/fontes-energia.htm>
- [2] R. Albadó, *Energia Eólica*, 1st ed., Teste, Ed. ArtLiber, 2002.
- [3] M. Lukosevicius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, pp. 127–149, 2009.
- [4] A. Araújo, Ferreira, "Um método para design e treinamento de reservatório computing aplicado à previsão de séries temporais," Ph.D. dissertation, Universidade de Federal de Pernambuco, 2011.
- [5] D. Verstraeten, "Reservoir computing: computation with dynamical systems," Ph.D. dissertation, Ghent University, 2009.
- [6] ONS. (2012) Operador nacional do sistema elétrico. Acessado em 11 de outubro de 2012. [Online]. Available: <http://www.ons.org.br/home/>
- [7] J. M. Valença, Silva, *Fundamentos das Redes Neurais*, T. Pereira, Ed. Livro Rápido, 2011.
- [8] d. O. C. S. dos Santos, Medeiros, "Um sistema híbrido baseado em rede neural e colônia de formigas," Master's thesis, Universidade de Pernambuco, 2010.
- [9] N. Juristo and A. Moreno, M., *Basics of Software Engineering Experimentation*. Kluwer Academic Publisher, 2001.