

Echo State Networks

Fabricio Breve
João Bertini

RNN – Tarefas

- Filtragem e predição
- Classificação de Padrões
- Modelagem de Seqüências Estocásticas
- Memória Associativa
- Compressão de Dados

RNN – Aplicações

- Telecomunicações
- Controle de usinas químicas
- Controle de motores e geradores
- Monitoramento de falhas, diagnósticos biomédicos e monitoramento
- Reconhecimento de fala
- Robótica e brinquedos
- Análise de dados de vídeo
- Interfaces homem-máquina

ESN - Definição

- Modelo recente de rede neural recorrente (RNR) com aprendizado de complexidade linear [Jaeger, Haas 2004].
- RNR – Rede neurais que usam saídas de neurônios da rede no tempo t , como entrada para outros neurônios no tempo $t+1$.

RNN – Algoritmos mais usados

- Backpropagation through time (BPTT)
 - Não garante convergência para mínimo global;
 - Extremamente lento;
 - Necessidade de tentativa e erro;
- Real time recurrent learning (RTRL)
 - Técnica de aprendizado com forte embasamento matemático;
 - Alto custo computacional;
 - Limitado a pequenas redes

ESN - Motivações

- RN Feedforward
 - difícil de treinar;
 - muitas vezes intratável;
 - não dinâmico;
- RN Recorrentes
 - boa performance;
 - dinâmico;
- **Echo State Networks:** Rede recorrente com aprendizado de complexidade linear.

ESN – Motivações

- Maioria dos modelos recorrentes requerem convergência a pontos estáveis [Hertz et al. 1991].
- Problemas
 - Necessidade de muitos pontos estáveis para armazenar informação (ex. 1024 ptos para 10 bits);
 - Tempo de convergência para pontos estáveis inviabiliza computação em tempo real;

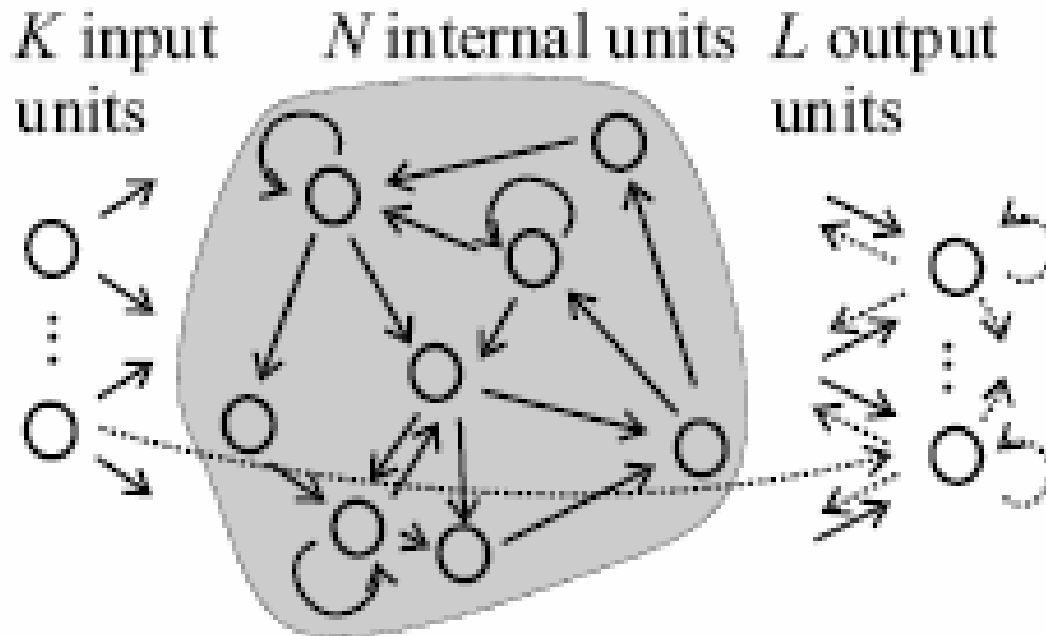
Propriedade de Echo State

- O estado de ativação $x(n)$ de uma rede neural recorrente é uma função do histórico de entradas $u(n), u(n-1), \dots$
- Ou seja, existe uma função E tal que:
$$x(n) = E(u(n), u(n-1), \dots)$$
- O estado $x(n)$ é visto como um eco das entradas anteriores.

ESN - Imposições

- Maior dos auto-valores de W tem que ser menor que 1;
- Conectividade de W esparsa (1-20%);

ESN - Estrutura



ESN - Funções de ativação

$$\mathbf{x}(n + 1) = f(\mathbf{W}_{in} \mathbf{u}(n + 1) + \mathbf{W}_x \mathbf{x}(n) + \mathbf{W}_{back} \mathbf{y}(n))$$

$$\mathbf{y}(n + 1) = f_{out}(\mathbf{W}_{out}(\mathbf{u}(n + 1) | \mathbf{x}(n + 1) | \mathbf{y}(n)))$$

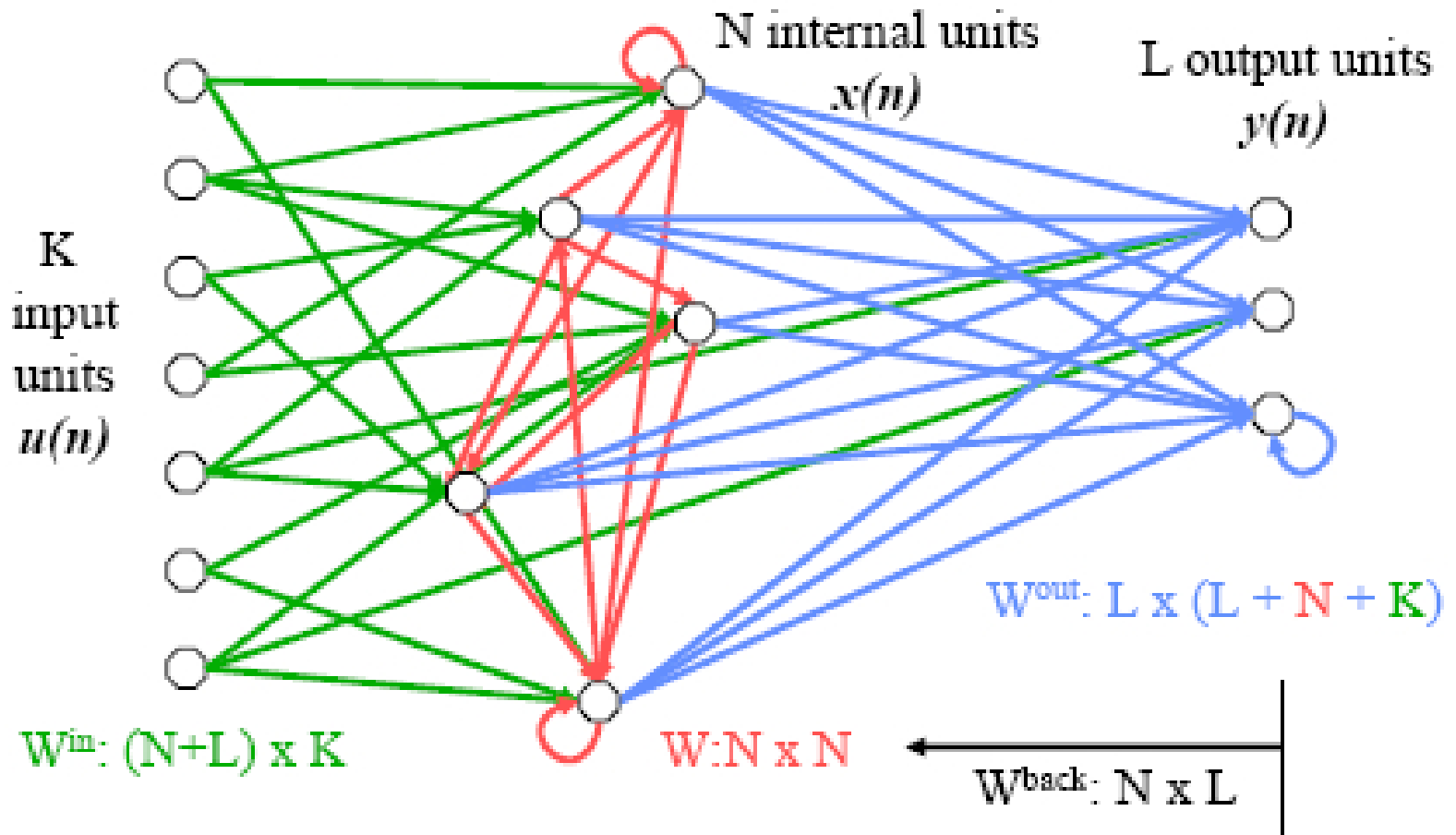
ESN

- Habilidade de treinar sem modificar pesos da rede.
- Somente os pesos do reservatório para a saída são calculados.
- Como não há dependências cíclicas entre os neurônios de saída, o treinamento torna-se linear

ESN - Reservatório

- O reservatório, possui grande número de neurônio (50 a 100);
- Neurônios são randomicamente conectados;
- O reservatório pode ser excitado pelo input ou pelo output;
- Conexões do reservatório não mudam durante treinamento;

ESN - Rede



Treinamento/Inicializações

- As inicializações de W , W_{in} e W_{out} são feitas por meio de alguma heurística, com a restrição de que W seja espessa e tenha raio espectral menor que 1.
- W_{in} e W_{out} :
 - Pesos pequenos geram redes com dinâmicas quase linear;
 - pesos grandes geram redes com dinâmica não-linear;

Treinamento

- Treine a rede apresentando a entrada $u(n)$ e forçando a saída $d(n-1)$, usando a equação:

$$x(n+1) = f(W_{in} u(n+1) + W_x(n) + W_{back} d(n))$$

- Para cada entrada, colete o estado $x(n) = (x_1(n), \dots, x_N(n))$ em uma linha da matriz M .
- Armazene também, o valor da função $f^{-1}d(n)$ em T .

Calculando os pesos de saída

- Executa-se o procedimento anterior por um tempo $T > T_0$.

- Então a matriz de pesos é definida por:

$$W_{\text{out}}^T = M^{-1} T$$

Referencias Bibliográficas

- **H. Jaeger (2002): Tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF and the "echo state network" approach.** GMD Report 159, German National Research Center for Information Technology, 2002 (48 pp.)
- **H. Jaeger(2001): The "echo state" approach to analysing and training recurrent neural networks.** GMD Report 148, German National Research Center for Information Technology, 2001 (43 pp.)

Referencias Bibliográficas

- H. Jaeger (2002): **Adaptive nonlinear system identification with echo state networks.** In Advances in Neural Information Processing Systems 15, S. Becker, S. Thrun, K. Obermayer (Eds), (MIT Press, Cambridge, MA, 2003) pp. 593-600
- Skowronski, M.D. Harris, J.G. Minimum mean squared error time series classification using an echo state network prediction model, In: Proceedings of the IEEE International Symposium in Circuits and Systems, ISCAS 2006.