Anais do IX Congresso Brasileiro de Redes Neurais / Inteligência Computacional (IX CBRN)
Ouro Preto 25-28 de Outubro de 2009

PROJETO DE UMA CALCULADORA ACIONADA POR VOZ PARA *SMARTPHONES* USANDO REDES NEURAIS AUTO-ORGANIZÁVEIS

Amauri H. S. Júnior*, Guilherme A. Barreto*, Antonio T. Varela†

*Av. Mister Hull, S/N
Universidade Federal do Ceará, Departamento de Engenharia de Teleinformática
Fortaleza, CE, Brasil

†Av. 13 de maio, 2081
Centro Federal de Educação Tecnológica, Instituto Tecnológico de Telecomunicações e Informática
Fortaleza, CE, Brasil

Emails: amauriholanda@ymail.com, guilherme@deti.ufc.br, themoteo@cefetce.br

Abstract— This paper is a comparative study of the main algorithms used to speech recognition, and assessment of the performance of approaches based on Kohonen network for use in embedded systems. The data set used in this work consists of 17 utterances corresponding to the digits and operations necessary for the development of a calculator powered by voice. The application was then developed on a Nokia N95 smartphone. The results indicate that the self-organizing networks can be applied in the task of interest successfully, and showed superior results comparing to other classical techniques regarding the recognition rates and computational cost, including possible training and adaptation to new words in the embeeded system.

Keywords— Speech Recognition, Self-Organizing Map, Embedded Systems.

Resumo— Neste artigo é apresentado um estudo comparativo entre os principais algoritmos utilizados para reconhecimento de voz, bem como a avaliação do desempenho de abordagens baseadas na rede de Kohonen para sua utilização em sistemas embarcados. O conjunto de dados utilizado neste trabalho consiste de 17 classes de elocuções, pronunciadas naturalmente, correspondendo aos dígitos e operações necessárias para o desenvolvimento de uma calculadora acionada pela voz. A aplicação foi então embarcada em um smartphone N95 da Nokia. Os resultados indicam que as redes auto-organizáveis podem ser aplicadas na tarefa de interesse com sucesso, e apresentaram resultados superiores aos de outras técnicas clássicas quanto à taxa de reconhecimento e custo computacional, incluindo possível treinamento e adaptação a novas palavras no próprio sistema embarcado.

Palavras-chave— Reconhecimento de Voz, Rede de Kohonen, Sistemas Embarcados.

1 Introdução

O mapa auto-organizável (SOM - *Self Organizing Map*) de Kohonen (Kohonen 1997) vem sendo aplicado em diversas áreas, tais como: reconhecimento de voz, robótica inteligente, análise de dados, entre outros. Porém, em aplicações em que se dispõe de recursos computacionais bastante limitados, como em sistemas embarcados, o uso da rede SOM na forma que foi proposta inicialmente torna-se impraticável (Koikkalainen 1994). Isto acontece devido ao custo computacional associado com treinamento e teste do mapa (Sagheer et al. 2006).

A busca pela redução computacional de algoritmos de quantização vetorial é antiga, um dos primeiros trabalhos com grande impacto foi desenvolvido por Friedman et al. (1977), e serviu de referência para o desenvolvimento de diversas técnicas, entre elas o Mapa Auto-Organizável estruturado em árvore (TS-SOM, *Tree-Structured Self-Organizing Map*) (Koikkalainen & Oja 1990) e a Quantização Vetorial estruturada em árvore (TSVQ, *Tree-Structured Vector Quantization*) (Buzo et al. 1980).

Dentre as várias áreas nas quais a rede SOM tem sido aplicada destaca-se a de reconhecimento de voz. A rede SOM tem sido utilizada tanto na codificação quanto na classificação da fala. Além disto, as abordagens mais recentes para reconhecimento de voz utilizam arquiteturas híbridas, tais como MLP (*MultiLayer Perceptrons*)-SOM, e SOM-HMM (*Hidden Markov Models*) (Kohonen 1997) no reconhecimento de fonemas, proporcionando o reconhecimento da fala contínua (Rabiner & Juang 2008).

Com o crescimento significativo da capacidade de processamento, os celulares atualmente são capazes de executar tarefas antes realizadas somente em computadores pessoais. No entanto, a interface de comunicação com o usuário ainda limita a usabilidade dos dispositivos móveis, visto que os teclados, forma convencional de interação, estão ficando cada vez menores. Diante disso, a interface de comunicação por voz tem se mostrado a forma mais natural e eficiente para atender a nova demanda de aplicações para dispositivos móveis.

Nesse contexto, o presente artigo descreve a aplicação de redes neurais auto-organizáveis no problema de reconhecimento de dígitos em sistemas embarcados, através do projeto de uma calculadora acionada pela voz. O sistema será embarcado no *smartphone* N95, pertencente à série S60 da Nokia. Dar-se-á ênfase ao estudo de técnicas de *software* para aceleração da rede SOM, na tarefa de recon-

hecimento de palavras isoladas e independentes do locutor.

O restante do artigo está organizado da seguinte forma. Na seção 2 são apresentas as principais arquiteturas avaliadas neste trabalho, e na seção seguinte, seção 3, algumas técnicas utilizadas para reduzir o custo computacional da rede de Kohonen são descritas. A seção 4 traz a avaliação do custo computacional da rede SOM, bem como o desempenho de redes auto organizáveis em comparação com outras técnicas clássicas de reconhecimento de voz. Além disto, essa seção apresenta resultados no dispositivo móvel utilizado na aplicação da calculadora acionada pela voz. A seção 5 conclui o artigo.

2 Arquiteturas Avaliadas

2.1 Mapa auto-organizável de Kohonen

O Mapa Auto-Organizável de Kohonen (SOM - *Self-Organizing Map*)(Kohonen 1997) é uma rede neural com aprendizagem não supervisionada que realiza uma projeção de um espaço de entrada geralmente contínuo e de alta dimensionalidade $\mathcal{X}$ em um espaço discreto de baixa dimensionalidade (mapa) $\mathcal{A}$ formado por M neurônios que são arranjados em uma topologia fixa, comumente uni ou bi-dimensional.

O mapa $i^*(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{A}$, definido pela matriz de pesos $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M), \mathbf{w}_i \in \mathcal{X}$, relaciona a cada vetor de entrada $\mathbf{x} \in \mathcal{X}$ um neurônio vencedor $i^* \in \mathcal{A}$ no mapa. Essa etapa chama-se busca pelo vencedor e é definida por

$$i^*(t) = \arg\min_{\forall i} \|\mathbf{x}(t) - \mathbf{w}_i(t)\| \qquad (1)$$

em que $\mathbf{x}(t) \in \Re^n$ denota o vetor de entrada atual, $\mathbf{w}_i(t) \in \Re^n$ é o vetor de pesos do neurônio i, e t simboliza a variável temporal associada com as iterações do algoritmo.

A outra etapa referente ao algoritmo é a atualização do neurônio vencedor e seus vizinhos:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t)h(i^*, i; t)[\mathbf{x}(t) - \mathbf{w}_i(t)] \quad (2)$$

em que $0 < \alpha(t) < 1$ é a taxa de aprendizagem, e $h(i^*, i; t)$ é uma função vizinhança, como por exemplo a função gaussiana:

$$h(i^*, i; t) = \exp\left(-\frac{\|\mathbf{r}_i(t) - \mathbf{r}_{i^*}(t)\|^2}{2\sigma^2(t)}\right) \qquad (3)$$

em que $\mathbf{r}_i(t)$ e $\mathbf{r}_{i^*}(t)$ são respectivamente, as posições dos neurônios i e i^* no mapa, e $\sigma(t) > 0$ define o raio de atuação da função vizinhança no tempo t. As variáveis $\alpha(t)$ e $\sigma(t)$ decaiam com o tempo para garantir convergência da rede. Além disso, no treinamento da rede, as operações definidas em (1) e (2) são repetidas para cada vetor de entrada durante ϵ épocas até que o um estado de ordenação global tenha sido alcançado.

A rede SOM tem sido amplamente utilizada em engenharia e tarefas de análise de dados, mas raramente utilizada em problemas de tempo real. A razão para isso está no custo computacional associado à rede. Na busca pelo neurônio vencedor, o algoritmo requer o cálculo de distâncias a todos os vetores protótipos da rede. A Tabela 1 exibe o número mínimo de operações (multiplicação, divisão, adição, subtração, comparação e exponenciação) do treinamento de uma rede SOM utilizando função vizinhança gaussiana, conforme a equação 3. Na Tabela 1, N é o número de vetores de entrada, M, o número de neurônios no mapa, n é a dimensão dos vetores de entrada, k representa a dimensão do mapa (uni ou bi - dimensional), e ϵ o número de épocas de treinamento.

Como pode ser visto na Tabela 1, a etapa de atualização é a que requer um maior número de operações. Observa-se ainda que tanto a busca quanto a atualização possuem complexidade linear em função do número de neurônios na rede. Assim, é comum referenciar a complexidade computacional da rede SOM como $O(M)$.

Como será visto adiante, não é necessário atualizar todos os protótipos da rede a cada novo padrão de entrada. Isto aconteceu aqui, pois a função vizinhança gaussiana é assintótica, e dessa forma mesmo os neurônios mais distantes são atualizados. Outros tipos de função vizinhança e adaptação da taxa de aprendizagem podem ser utilizadas para reduzir o custo computacional da rede. Com isto, a etapa de busca pelo vencedor torna-se computacionalmente dominante e grande parte das técnicas para aceleração computacional do algoritmo SOM se concentram na otimização dessa busca.

2.2 Mapa auto-organizável estruturado em árvore

O mapa auto-organizável estruturado em árvore, ou simplesmente TS-SOM (*Tree-Structured Self-Organizing Map*), foi inicialmente proposto por Koikkalainen & Oja (1990) como uma alternativa rápida ao algoritmo de treinamento/teste da rede SOM, com complexidade computacional $O(logM)$. Em seguida, Koikkalainen (1994) incorporou melhorias e considerações importantes ao algoritmo.

O TS-SOM é constituído de várias redes SOM de diferentes resoluções, em que os nós (neurônios) de uma mesma camada são conectados lateralmente. Além disso, existem conexões hierárquicas entre nós de diferentes camadas. Normalmente, cada

Tabela 1: Número de operações da rede SOM.

	busca	atualização	adapt	total
multi.	$NnM\epsilon$	$NM\epsilon(4+k+n)$	2ϵ	$\epsilon[2+NM(2n+k+4)]$
divi.	-	$NM\epsilon$	ϵ	$\epsilon(NM+1)$
adi.	$NM\epsilon(n-1)$	$NM\epsilon(k-1+n)$	-	$NM\epsilon(2n+k-2)$
sub.	$NnM\epsilon$	$NM\epsilon(k+n)$	-	$NM\epsilon(2n+k)$
comp.	$(M-1)N\epsilon$	-	-	$N\epsilon(M-1)$
exp.	-	$NM\epsilon$	2ϵ	$\epsilon(2+NM)$
total	$N\epsilon(3Mn-1)$	$MN\epsilon(3n+3k+5)$	5ϵ	

nó está interligado a 2^D neurônios na próxima camada, em que D é a dimensão da rede SOM.

No treinamento da rede, os neurônios são adaptados camada por camada. Após o treinamento de uma camada, todos os neurônios dela são "congelados", ou seja, seus pesos sinápticos não são alterados. A seguir mostram-se os principais passos do algoritmo:

1. Inicializar o nó raiz da árvore com valores aleatórios.
2. Treinar a camada atualmente adaptável da árvore até que um critério de convergência seja alcançado:
   (a) Escolher aleatoriamente um vetor de entrada do conjunto de dados disponível para treinamento.
   (b) Encontrar o neurônio vencedor i^*, na camada adaptável;
   (c) Atualizar os pesos de i^* e seus vizinhos imediatos em direção ao vetor de entrada selecionado.
3. Inicializar os pesos dos neurônios da próxima camada com os valores dos pesos dos seus pais e treinar a nova camada (voltar para passo 2).

A etapa essencial e que mais difere do SOM original é a busca pelo neurônio vencedor. Na busca em árvore comum a busca vai de um nó da árvore ao seu filho mais similar ao padrão de entrada - filho vencedor. Já no TS-SOM, o conjunto de busca inclui os filhos do neurônio vencedor e de seus vizinhos imediatos. A inclusão dos filhos dos vizinhos imediatos ao neurônio vencedor da camada anterior à adaptativa é bastante econômica computacionalmente, pois não depende do número de neurônios na camada, além de ser vital para o funcionamento do algoritmo (Koikkalainen & Oja 1990).

A atualização no TS-SOM é bastante similar ao SOM original, de modo que o neurônio vencedor i^*, e seus vizinhos topológicos imediatos (neurônios da mesma camada que estão conectados ao vencedor), são ajustados em direção ao vetor de entrada.

3 Técnicas de Redução do Custo Computacional da rede SOM

Nesta seção são apresentados algumas técnicas para redução do número de operações no treinamento e teste da rede SOM.

3.1 Busca com Distância Parcial

Buscando reduzir o número de operações nos algoritmos de quantização vetorial, Bei & Gray (1985) propuseram o algoritmo busca com distância parcial, ou PDS (*Partial Distance Search*). O método proposto reduz o custo computacional no cálculo da distorção ou dissimilaridade entre vetores pela medade ou mais, sem perda de eficiência (Bei & Gray 1985).

O algoritmo é uma forma simples de reduzir o número de operações de multiplicação, adição e subtração na busca pelo neurônio vencedor. Se a distância parcial quadrática entre um vetor de entrada e o vetor de um protótipo excede a distância quadrática total para o protótipo mais próximo encontrado até o instante atual na busca, este protótipo pode ser descartado. Isto é, pode-se descartar o protótipo em que a distorção ou distância acumulada nas primeiras j ($j < n$) amostras é maior que a menor distância encontrada na busca.

Na busca com descarte antecipado de protótipos, o tempo adicional gasto para a comparação ($d > d_{min}$) após o cálculo da distorção em cada dimensão é, em média, menor que o tempo gasto na busca completa pelo vencedor (Niskanen et al. 2002), (Bei & Gray 1985).

3.2 Busca com Atalho

Kohonen (1997) propôs a busca com atalho (*Shortcut Winner Search*, em inglês) com o intuito de reduzir o custo computacional na busca pelo vencedor na rede SOM. A solução encontrada por Kohonen é similar à aceleração do TS SOM proposta por Koikkalainen (1994). No entanto, a busca com atalho pode ser aplicada a qualquer arquitetura

SOM, e não necessita de divisão em camadas ou níveis hierárquicos.

Após algumas iterações do treinamento, de forma que o mapa esteja suavemente ordenado, o tamanho da função vizinhança já está pequeno, a taxa de aprendizagem está menor, e portanto o número de correções no mapa é pequeno. Com isto, a probabilidade de o neurônio vencedor para um padrão $\mathbf{x}$ ser ou estar na vizinhança do neurônio vencedor para o mesmo padrão $\mathbf{x}$ na época anterior é alta.

Diante disso, Kohonen (1997) então sugere armazenar uma referência, ou ponteiro, relacionando um vetor $\mathbf{x}$ ao protótipo vencedor i^*, na iteração t. Na iteração $t+1$ a busca pode ser feita na vizinhança imediata do neurônio i^*, e somente se um protótipo mais próximo de $\mathbf{x}$ for encontrado nessa vizinhança a busca continua na vizinhança desse novo neurônio e assim sucessivamente, até que o vencedor esteja no centro do domínio da busca. Após o vencedor ter sido identificado, a referência ao vetor $\mathbf{x}$ é então atualizada. É importante ressaltar que a cada nova vizinhança a ser procurada, somente os protótipos que não foram testados anteriormente precisam ser examinados.

3.3 Evitar função raiz quadrada

Para calcular a distância euclidiana entre dois vetores uma função raiz quadrada é utilizada. No entanto, quando somente as relações entre as distâncias realmente importam, essa raiz não precisa ser calculada. Esta situação é exatamente o caso da busca pelo vencedor. Assim, no cálculo de distâncias entre dois vetores $\mathbf{x}$ e $\mathbf{y} \in \Re^n$, deve ser utilizado o quadrado da distância $d(\mathbf{x}, \mathbf{y})^2$.

3.4 Funções vizinhanças

Kohonen (1997) define uma vizinhança do tipo retangular ou bolha, ou seja, dado um raio $R(t)$ do neurônio vencedor no passo t, $N_{i^*}(t)$ o conjunto de neurônios tal que $\| \mathbf{r}_{i^*}(t) - \mathbf{r}_i(t) \| < R(t)$, então $h(i^*, i; t) = 1$ para todos os neurônios $i \in N_{i^*}(t)$ e $h(i^*, i; t) = 0$ caso contrário.

Uma outra possível função vizinhança é a gaussiana truncada. Neste caso, são calculadas as funções gaussianas para todos os neurônios, mas somente os neurônios que estiverem acima de um limiar são atualizados.

A grande vantagem computacional na utilização dessas abordagens é que somente um pequeno número de neurônios precisa ter seus pesos alterados para um determinado padrão $\mathbf{x}$, e além disso, não há cálculo de exponenciais, que são computacionalmente mais custosas do que somas, subtrações e multiplicações.

4 Simulações e Resultados

Com o intuito de avaliar o uso da rede SOM para reconhecimento de voz e criar um banco de dados de palavras foi desenvolvida a aplicação mostrada na Figura 1.

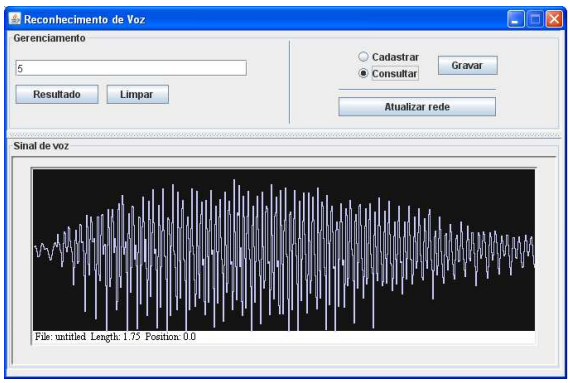

Figura 1: Interface Gráfica do Sistema de Reconhecimento de Voz.

O aplicativo foi desenvolvido em Java e banco de dados MySQL. A aplicação possui quatro botões: *Limpar*, *Resultado*, *Gravar* e *Atualizar Rede*; um campo de texto; dois *checkboxes* - *Cadastrar* e *Consultar*, e um quadro para exibir o sinal de voz, permitindo uma avaliação qualitativa dos sinais capturados.

A aplicação permite o reconhecimento de palavras isoladas. Com o *checkbox Cadastrar* selecionado, a voz gravada clicando-se no botão *Gravar* é armazenada no banco MySQL. Além disto, é necessário que o usuário escreva a palavra que será cadastrada no campo de texto, pois esta será também armazenada. Para testar a aplicação, o usuário deve selecionar a opção *Consultar* e *Gravar* sua elocução.

Na criação do conjunto de dados foi capturada a voz de 10 pessoas, pronunciando cada palavra 3 vezes, variando-se a distância ao microfone. As palavras enunciadas foram: os dígitos de 0 a 9, *vezes*, *dividido*, *limpar*, *voltar*, *resultado*, *mais*, e *menos*. Quanto ao pré-processamento, a aplicação desenvolvida utiliza taxa de amostragem de 8 KHz, 8 bits na quantização das amostras do sinal e pré-ênfase. No recorte das palavras foi utilizado o algoritmo de Rabiner & Sambur (1975), com os limiares iguais a $A = 42$, $B = 32$ e $C = 22$. Além disto, 11 coeficientes Cepstrais obtidos a partir de coeficientes LPC (Deller et al. 2000), extraídos de cada quadro de 20 ms com superposição de 25%, representaram as características ou atributos de cada elocução. O

transdutor utilizado foi um microfone para computadores pessoais da marca XPC. Vale ressaltar que as palavras foram pronunciadas sem qualquer tipo de controle, ou seja, de forma espontânea, em sala fechada, com ruído proveniente de condicionadores de ar.

Para efeito comparativo, as redes SOM e TS-SOM 1D foram comparadas com técnicas clássicas de reconhecimento de padrões, como a rede neural MLP (*Multi-Layer Perceptron*)(Bishop 1995), Classificador do Vizinho Mais Próximo utilizando K-Médias como quantizador vetorial (Rabiner & Juang 1993), e DTW (*Dynamic Time Warping*) (Sakoe & Chiba 1978).

A abordagem utilizada para os algoritmos TS-SOM, SOM e K-Médias foi a de quantização vetorial encontrada em Rabiner & Juang (1993), em que é associada uma rede para cada classe de elocução. Nos experimentos com o TS-SOM foi utilizado uma rede com 10 camadas, 2 filhos, resultado em 256 protótipos. Nas redes SOM e K-Médias utilizou-se 500 épocas de treinamento, e no TS-SOM, 5000, uma vez que a forma do treinamento é diferente. Todas as redes possuem 256 protótipos. Estes valores foram escolhidos com base nos testes realizados, os quais indicaram que redes com menor número de protótipos possuiam uma menor taxa de classificação. Todas as redes tiveram seus pesos iniciados aleatoriamente entre -1 e 1.

Com relação às redes neurais com treinamento supervisionado, tais como as redes MLP e SVM, foram realizados experimentos com diversas métodos de normalização do sinal de voz, de forma que o número de atributos por palavra fosse fixo, independente do tamanho do sinal. Com a quantização vetorial dos sinais obteve-se as melhores taxas de acerto. Assim, para esta tarefa foi utilizado uma rede TS-SOM com 5 camadas e 2 neurônios filho, resultando em 16 protótipos. Além disto, a rede MLP utilizada possui 80 neurônios ocultos, e uma única camada oculta.

A avaliação dos classificadores foi realizada utilizando 80% dos dados para treinamento (oito locutores) e 20% (dois locutores) para teste. Para cada classificador foram coletadas as taxas de acerto Média, Mínima, Máxima, Desvio Padrão, tempo médio da etapa de treinamento e tempo médio na classificação de um padrão (elocução), para um conjunto de 10 simulações independentes. Os resultados são mostrados na Tabela 2.

Considerando a taxa de acerto média observa-se que as metodologias baseadas na rede SOM obtiveram os melhores resultados, com taxa de acerto média por volta de 85%. O piores resultados médio foram das rede K-Médias e MLP, embora o tempo de classificação de um padrão com a rede MLP seja mais de mil vezes mais rápida que o dos demais algoritmos. Além disto, todos os algoritmos obtiveram a mesma ordem no desvio padrão das taxas de acerto.

Ainda observando a Tabela 2 nota-se que os algoritmos SOM(*Atalho+Retangular*) e SOM(*PDS+Retangular*) conseguiram reduzir o tempo de treinamento em cerca de 20%. Com relação ao tempo de teste, já era esperado que não houvesse redução com a busca com atalho, uma vez que ela só é aplicada no treinamento. Assim, a redução significativa aconteceu com as abordagens que utilizaram PDS, resultando em redução de aproximadamente 53% no tempo de classificação ou teste.

Analisando conjuntamente os diversos aspectos, as redes SOM(*PDS+Retangular*), TS-SOM, TS-SOM(PDS) são as mais adequadas para a aplicação embarcada quando comparadas com as demais redes e algoritmos avaliados, pois obtiveram as melhores taxas de acerto com o menor tempo de treinamento ou teste. Caso a aplicação exija o treinamento *online*, ou seja, no próprio sistema embarcado, a rede TS-SOM(PDS) possui vantagem pois seu treinamento dura em média 5 s.

Visando validar o uso de redes neurais auto-organizáveis em sistemas embarcados, foi desenvolvida uma calculadora acionada pela voz no *smartphone* N95 da Nokia. A interface da aplicação é mostrada na Figura 2.

(a) Tela de menu no emulador da SUN Microsystems.

(b) Aplicação no N95.

Figura 2: Calculadora acionada pela voz.

A aplicação foi desenvolvida utilizando-se um *framework* Java chamado JME (*Java Micro Edition*), criado para o suporte à aplicações embarcadas. A captura de voz foi realizada através da API (*Application Programming Interface*) *Mobile Media API* (JSR-135). Em decorrência do estudo realizado, foi escolhida a rede TS-SOM para o reconhecimento das palavras. Com isto, a rede pode inclusive ser treinada no dispositivo móvel. Além disto, foram utilizados 11 coeficientes cepstrais na extração de características, e todo o pré-processamento

Tabela 2: Desempenho dos classificadores.

	Média (%)	Máx. (%)	Mín. (%)	Desvio (%)	Treino (ms)	Teste (ms)
SOM(*Clássico*)	86,66	91,35	82,71	5,37	1.537.159,04	8,556
SOM(*PDS+Retangular*)	81,48	82,71	80,24	1,00	1.248.630,51	3,971
SOM(*PDS+Truncada*)	84,56	88,88	81,48	3,10	1.446.252,01	4,158
SOM(*Atalho+Retangular*)	85,18	87,65	81,48	3,02	1.240.714,55	8,462
TS-SOM	87,16	90,12	83,95	2,84	15.878,47	27,950
TS-SOM(*PDS*)	86,66	90,12	82,71	2,94	5.695,35	9,556
MLP	69,01	76,54	59,25	4,42	196.521,94	0,085
K-Médias	61,95	74,07	44,44	8,24	89.041,84	10,792
DTW	75,30	86,41	64,19	9,07	38.279,87	1.738,810

conforme os testes *offline* realizados anteriormente neste trabalho.

A aplicação possui um menu principal, em que se pode observar as elocuções já cadastradas e cadastrar uma nova. Na utilização da calculadora, o usuário observa uma barra de progresso, correspondendo ao tempo em que ele deve pronunciar uma palavra. Ao final da barra de progresso, o conteúdo dito é processado e o digito ou operação reconhecida é mostrada no campo de texto da aplicação. Caso nada seja pronunciado, a barra inicia novamente sua execução. Para isto, foi utilizado um limiar experimental com base na energia do sinal capturado.

5 Conclusões

Este trabalho apresentou uma avaliação das principais técnicas de reconhecimento de fala espontânea independente do locutor. O trabalho se concentrou na utilização de abordagens baseadas na rede auto-organizável de Kohonen e seu uso em sistemas embarcados, mais especificamente em um dispositivo móvel da Nokia.

A partir disso, foram desenvolvidas aplicações de reconhecimento de voz. A primeira, *offline*, possibilitou a formação de um conjunto de elocuções, e uma avaliação estatística de taxas de reconhecimento e tempo de processamento dos algoritmos. Com isso, as redes baseadas no algoritmo auto-organizável de Kohonen apresentaram bons resultados em comparação com outros métodos clássicos. A segunda aplicação consistiu do desenvolvimento de uma calculadora acionada pela voz em um dispositivo móvel, buscando validar o estudo apresentado neste trabalho. Além disso, fica notória a necessidade de técnicas para aceleração da rede SOM para sua utilização em dispositivos com baixo poder de processamento.

Agradecimentos

À CAPES pelo apoio financeiro a este trabalho.

Referências

Bei, C.-D. & Gray, R. (1985). An improvement of the minimum distortion encoding algorithm for vector quantization, *IEEE Transactions on Communications* **33**(10): 1132–1133.

Bishop, C. M. (1995). *Neural Networks Pattern Recognition*, Oxford University Press.

Buzo, A., Gray, A., J., Gray, R. & Markel, J. (1980). Speech coding based upon vector quantization, *IEEE Transactions on Acoustics, Speech and Signal Processing* **28**(5): 562–574.

Deller, J. R., Hansen, J. H. L. & Proakis, J. G. (2000). *Discrete-Time Processing of Speech Signals*, John Wiley & Sons.

Friedman, J. H., Bentley, J. L. & Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time, *ACM Trans. Math. Softw.* **3**(3): 209–226.

Kohonen, T. K. (1997). *Self-Organizing Maps*, 2nd extended edn, Springer-Verlag, Berlin, Heidelberg.

Koikkalainen, P. (1994). Progress with the tree-structured self-organizing map, *European Conference on Artificial Intelligence (ECAI'94)*, pp. 211–215.

Koikkalainen, P. & Oja, E. (1990). Self-organizing hierarchical feature maps, *International Joint Conference on Neural Networks (IJCNN'90)*, pp. 279–284 vol.2.

Niskanen, M., Kauppinen, H. & Silvan, O. (2002). Real-time aspects of som-based visual surface inspection., *Proceedings SPIE Machine Vision Applications in Industrial Inspection*.

Rabiner, L. & Juang, B.-H. (1993). *Fundamentals of speech recognition*, Prentice-Hall International.

Rabiner, L. & Juang, J. H. (2008). Historical perspective of the field of ASR/NLU, *in* J. Benesty, M. M. Sondhi & Y. Huang (eds), *Springer Handbook of Speech Processing*, Springer, pp. 301–325.

Rabiner, L. & Sambur, M. R. (1975). An algorithm for determining the endpoints of isolated utterances, *Bell System Technical Jornal* **54**: 297–315.

Sagheer, A., Tsuruta, N., Maeda, S., Taniguchi, R.-I. & Arita, D. (2006). Fast competition approach using self organizing map for lip-reading applications, *International Joint Conference on Neural Networks (IJCNN '06)*, pp. 3775–3782.

Sakoe, H. & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition, *IEEE Transactions on Acoustics, Speech and Signal Processing* **26**(1): 43–49.