

Trabalho de Cognição Visual

Sumário

O principal objetivo do trabalho é implementar um computador neural (CN) capaz de executar programas especificados como sequências de padrões de ativação de camadas neurais. Outro objetivo importante deste trabalho é avaliar o desempenho do CN no controle de um robô autônomo, sendo este desempenho medido em termos da capacidade do CN de viabilizar a proceduralização de atividades complexas do robô autônomo (transformação de processos controlados em processos automáticos, pag. 112 do livro).

Arquitetural do Computador Neural

A arquitetura básica do CN é apresentada na Figura 1, abaixo.

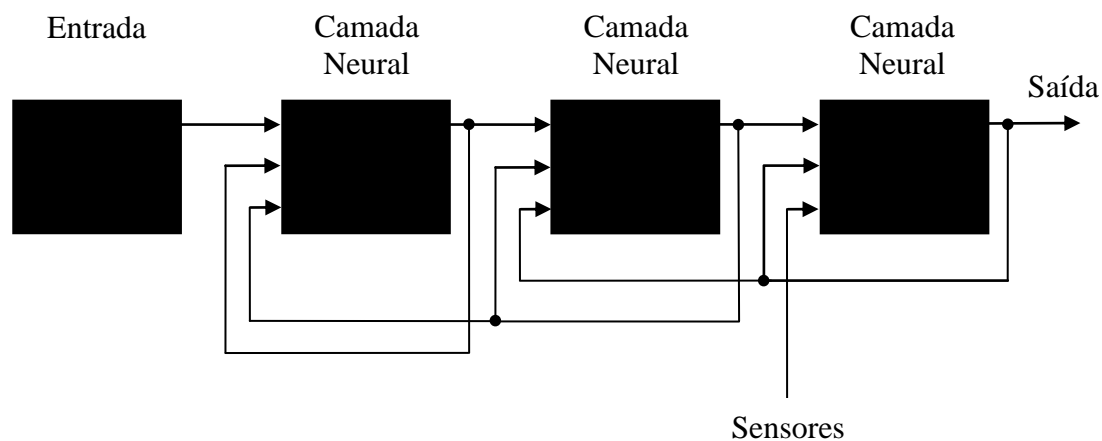


Figura 1: Computador Neural (CN)

Como a Figura 1 mostra, o CN é composto por quatro partes fundamentais: (i) a entrada principal (Entrada); (ii) 3 Camadas Neurais; (iii) uma entrada para sensores (Sensores); e (iv) uma saída (Saída). Todas estas partes são receptáculos de imagens bidimensionais (Entrada e Camadas Neurais), trazem imagens bidimensionais (Sensores), ou disponibilizam imagens bidimensionais (Saída).

As Camadas Neurais do CN são redes neurais sem peso do tipo VG-RAM bidimensionais com $n \times m$ neurônios. Cada neurônio destas Camadas Neurais possui um conjunto de sinapses que o interconectam com outras Camadas Neurais ou outros elementos da arquitetura do CN como mostrado pelas setas na Figura 1. O padrão de interconexão da arquitetura do CN forma uma estrutura hierárquica de camadas neurais, que permite denominá-las como mostrado na Figura 2.

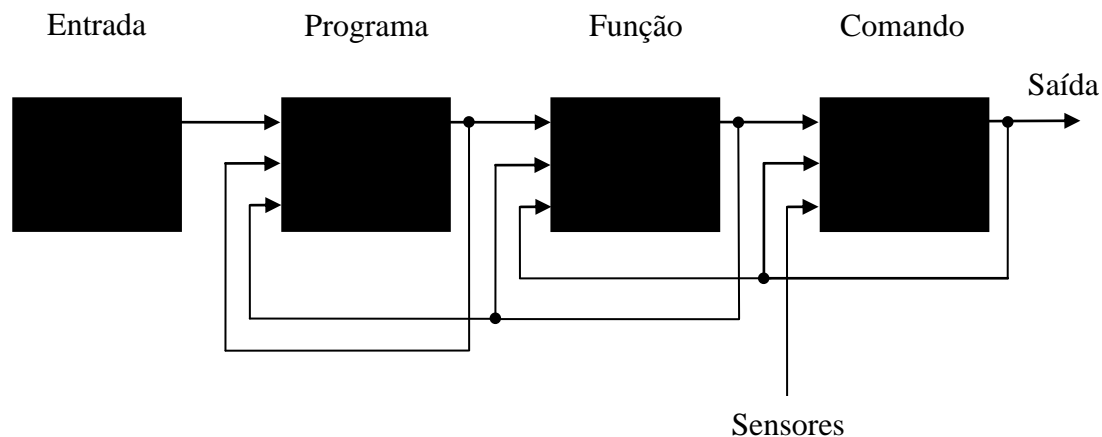


Figura 2: Arquitetura neural hierárquica do CN

As denominações das Camadas Neurais mostrada na Figura 2 indicam o papel esperado de cada Camada. A Camada Neural Programa, no nível hierárquico que ocupa, aprende programas demandados por Entrada e representados como uma sequência de funções. A Camada Neural Função aprende funções, representadas como sequências de comandos. Por fim, a Camada Neural Comando aprende comandos, representados por sequências de ações sobre a Saída. A entrada Sensores permite ao CN monitorar a execução dos comandos e, assim, a execução do programa como um todo.

Estruturalmente as Camadas Neurais do CN não diferem – todas elas são capazes de memorizar (aprender) relações entre imagens de entrada e imagens de saída. Isto é, dado um conjunto de imagens de entrada e uma imagem de saída, uma Camada Neural pode ser treinada de modo a, quando um conjunto de imagens de entrada previamente aprendido for reapresentado (ou uma versão aproximada dele), disponibilizar a saída correspondente (aprendida). Contudo, do ponto de vista lógico, elas diferem, pois, como discutido acima, as Camadas Neurais mais à esquerda ocupam um nível hierárquico mais alto que as mais à direita.

Redes Neurais Sem Peso VG-RAM

Redes Neurais Sem-Peso (RNSP) são redes neurais artificiais também conhecidas como redes neurais *Booleanas* ou *n-uplas* Classificadoras, ou seja, redes neurais artificiais cujos neurônios guardam conhecimento em memórias do tipo RAM (*Random Access Memories*). Estes neurônios artificiais operam em função dos valores binários que são apresentados em suas entradas e usam RAMs como tabelas-verdade: as *sinapses* de cada neurônio coletam valores de um vetor de *bits* da entrada da RNSP, que são usados como um endereço de memória RAM, onde é armazenado o valor de saída do neurônio. O treinamento de um padrão em uma RNSP pode ser feito em um único passo e consiste basicamente em armazenar a saída desejada nos endereços associados ao respectivo vetor de entrada apresentado aos neurônios.

Apesar da notável simplicidade, as RNSP são muito efetivas como ferramentas de reconhecimento de padrões, oferecendo treinamento rápido e de fácil implementação. No entanto, se o tamanho da entrada da RNSP for muito grande, o tamanho das RAMs dos neurônios torna proibitivo a sua implementação, já que deve possuir um número de

entradas igual a 2 elevado à potência deste tamanho. As redes VG-RAM (*Virtual Generalizing RAMs*) são redes baseadas em RAMs que requerem somente capacidade de memória suficiente para guardar dados relacionados ao conjunto de treinamento. Nos neurônios deste tipo de rede, são armazenados os pares de entrada-saída apresentados durante a fase de treinamento, e não somente as saídas. Na fase de teste, todas entradas dos pares entrada-saída aprendidos e armazenados nas memórias dos neurônios VG-RAM são comparadas com a entrada da rede. A saída de cada neurônio VG-RAM é determinada pela saída do par de entrada-saída (já aprendido) mais “próximo”. A medida de proximidade adotada é a *Distância de Hamming*, i.e., o número de bits diferentes entre dois vetores de bits de igual tamanho. No caso de mais de um par de entrada-saída apresentarem a mesma distância, a escolha é feita randômicamente entre estes pares.

A Figura 3 ilustra a tabela-verdade de um neurônio VG-RAM com três sinapses: X_1 , X_2 e X_3 . Esta tabela-verdade contém três posições, cada posição contendo um par entrada-saída armazenado durante a fase de treinamento: entradas #1, #2 e #3. Durante a fase de teste, quando um vetor de entrada é apresentado à RNSP, o algoritmo de teste VG-RAM calcula a distância entre este vetor de entrada e cada entrada já armazenada na tabela-verdade. No exemplo da Figura 1, a distância de Hamming entre o vetor de entrada (*input*) e #1 é dois (2), porque ambos os bits X_2 e X_3 não apresentam pareamento/semelhança com o vetor de entrada. A distância de #2 é um (1) pois X_1 é o único bit diferente. A distância da entrada #3 é três (3) pois todos os bits são diferentes. Para este vetor de entrada, é produzido o resultado *label2*, que é o valor armazenado em #2, como a saída do neurônio Y .

lookup table	X_1	X_2	X_3	Y
entry #1	1	1	0	label 1
entry #2	0	0	1	label 2
entry #3	0	1	0	label 3
input	1	0	1	label 2

Figura 3: Tabela-verdade de um neurônio VG-RAM.

RNSP podem ser usadas para o reconhecimento de imagens. Em problemas de reconhecimento de imagens baseados em aprendizado de máquina, deseja-se saber qual padrão deve ser atribuído a uma imagem dado um conjunto de pares imagem-padrão associado previamente aprendido pelo sistema. Para se reconhecer imagens automaticamente usando RNSP do tipo VG-RAM, uma Camada Neural de neurônios VG-RAM pode ser utilizada. Cada um dos neurônios desta Camada Neural possui um conjunto de sinapses que são conectadas de modo randômico à imagem de entrada e captura desta um padrão de bits.

Durante a fase de treinamento, cada imagem pertencente ao conjunto de pares imagem-padrão de treinamento é apresentada à entrada da Camada Neural e as saídas dos seus neurônios são conectadas ao padrão associado à imagem. Os neurônios da Camada Neural são treinados de forma a externar este padrão (uma imagem) dada a imagem de entrada. Após o treinamento, a operação de uma Camada Neural ocorre em ciclos contínuos de três fases executadas por cada um de seus neurônios: (i) leitura das entradas; (ii) busca na memória pelo padrão de bits de entrada aprendido mais próximo

ao padrão de bits colhido da imagem de entrada corrente; e (iii) disponibilização na saída do pixel do padrão associado à imagem mais próxima aprendida.

É importante notar que, no NC (Figura 1), uma camada neural possui 3 imagens de entrada (uma imagem vinda da camada mais acima na hierarquia, uma imagem vinda da própria camada neural, e uma imagem vinda da camada mais abaixo na hierarquia). No entanto, estas 3 imagens podem ser vistas como sendo uma só, composta pela combinação das 3.

Exemplo de Programação e Execução de um Programa no CN

Para compreender como o CN funciona vamos examinar um exemplo de programação do mesmo para que ele aprenda a contar de 0 a 9 três vezes. Isto é, vamos mostrar como ele pode ser usado para “proceduralizar” a atividade de contar de 0 a 9 três vezes. Vamos chamar este programa de “Contar de 0 a 9 três vezes” e implementá-lo de modo *top-down*. A Figura 4 mostra o primeiro passo (i) de especificação do programa.

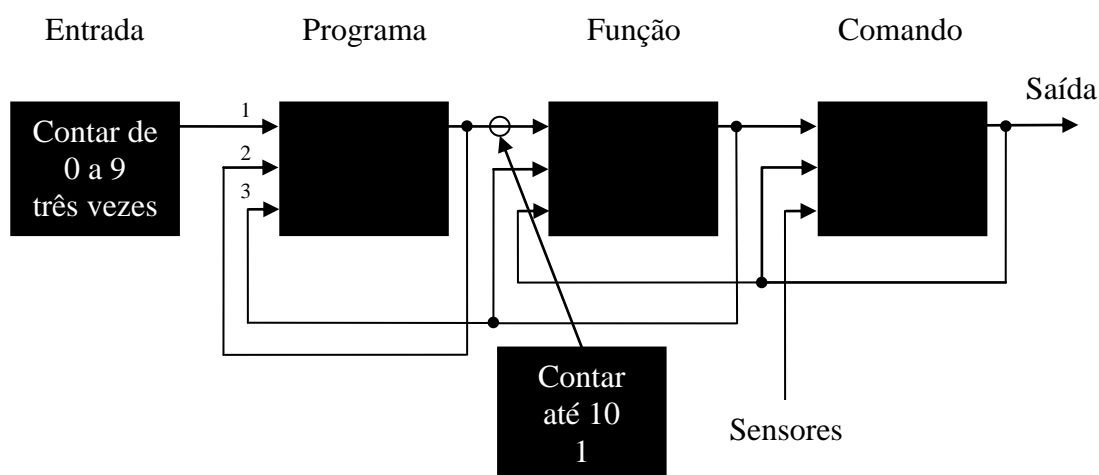


Figura 4: Passo (i) do programa “Contar de 0 a 9 três vezes”

Como apresentado na Figura 4, no passo (i) do programa “Contar de 0 a 9 três vezes”, especificamos na Entrada o nome do programa a ser executado e indicamos qual deve ser a saída da Camada Neural Programa. Na verdade, especificamos na Entrada apenas uma imagem que, por conveniência, pode ser interpretada por humanos como a mensagem “Contar de 0 a 9 três vezes”; e indicamos à Camada Neural Programa apenas uma imagem que deve ser aprendida, que também por conveniência pode ser lida por humanos como “Contar até 10 1”, ou seja, parte 1 de contar de 0 a 9 três vezes.

Especificados os itens que compõem o passo (i) do programa, podemos então treinar a Camada Neural Programa para gerar como saída a imagem “Contar até 10 1” quando observar nas suas entradas 1, 2 e 3 as imagens “Contar de 0 a 9 três vezes”, vazio (todos os pixels iguais a 0) e vazio, respectivamente.

No passo (ii) do programa “Contar de 0 a 9 três vezes”, mostrado na Figura 5, especificamos novamente na Entrada o nome do programa e indicamos qual deve ser a saída da Camada Neural Programa; além disso, especificamos o conteúdo da Camada Neural Programa, “Contar até 10 1”. Especificados os itens que compõem o passo (ii)

do programa, podemos então treinar a Camada Neural Programa para gerar como saída a imagem “Contar até 10 1” quando observar nas suas entradas 1, 2 e 3 as imagens “Contar de 0 a 9 três vezes”, “Contar até 10 1” e vazio, respectivamente. Ou seja, até este ponto da programação do CN, a Camada Neural Programa foi treinada para gerar a saída “Contar até 10 1” em duas circunstâncias: uma quando suas entradas 1, 2 e 3 contiverem as imagens “Contar de 0 a 9 três vezes”, vazio e vazio, respectivamente; e outra quando suas entradas 1, 2 e 3 contiverem as imagens “Contar de 0 a 9 três vezes”, “Contar até 10 1” e vazio, respectivamente.

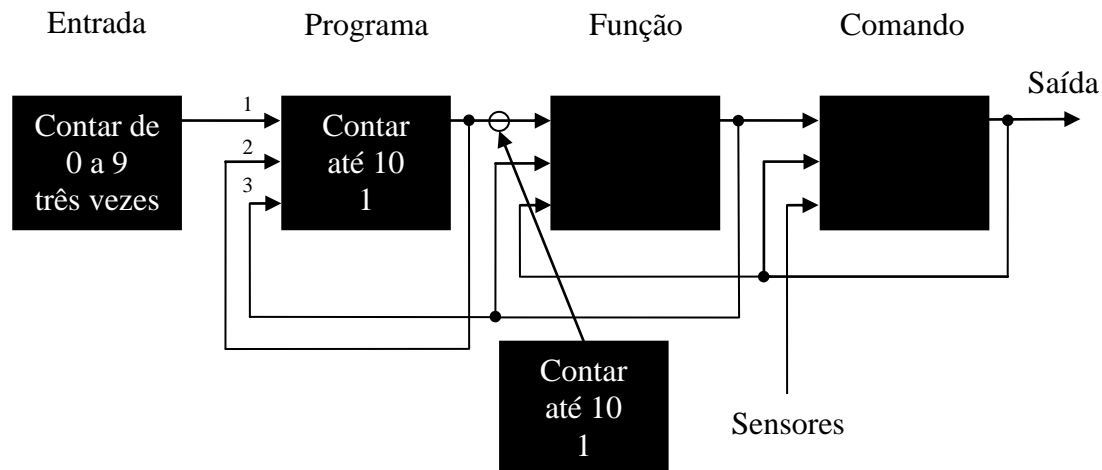


Figura 5: Passo (ii) do programa “Contar de 0 a 9 três vezes”

Como mencionado na seção anterior, após o treinamento, a operação de uma Camada Neural do CN ocorre em ciclos contínuos de três fases: leitura das entradas, busca na memória pelo padrão de entrada aprendido mais próximo ao padrão de entrada corrente, e disponibilização na saída dos pixels associados a este padrão de entrada aprendido. Com o que programamos até aqui, se o CN for inicializado com suas camadas neurais vazias e com a Entrada “Contar de 0 a 9 três vezes”, após o primeiro ciclo teremos na Camada Neural Programa a imagem “Contar até 10 1”. Em ciclos posteriores, dado o que foi programado até este ponto no CN, a Camada Programa não se alteraria.

No passo (iii) de nosso programa, indicamos qual deve ser a saída da Camada Neural Função – “Depois de vazio” – quando esta receber na entrada 1 a imagem “Contar até 10 1”, e nas entradas 2 e 3 a imagem vazio (ver Figura 6).

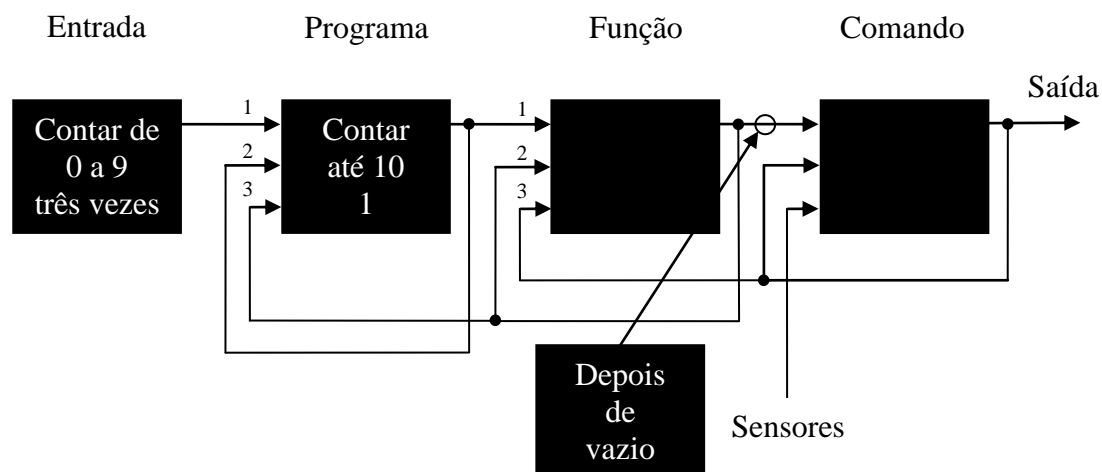


Figura 6: Passo (iii) do programa “Contar de 0 a 9 três vezes”

Especificado o item (iii) do programa, podemos então treinar a Camada Neural Função para gerar como saída a imagem “Depois de vazio” quando observar nas suas entradas 1, 2 e 3 as imagens “Contar até 10 1”, vazio e vazio, respectivamente.

No passo (iv) de nosso programa, treinamos a Camada Neural Função para gerar a saída “Depois de vazio” quando observar nas suas entradas 1, 2 e 3 as imagens “Contar até 10 1”, “Depois de vazio” e vazio, respectivamente. Treinamos também a Camada Neural Programa novamente para gerar como saída a imagem “Contar até 10 1” quando observar, agora, nas suas entradas 1, 2 e 3, as imagens “Contar de 0 a 9 três vezes”, “Contar até 10 1” e “Depois de vazio”, respectivamente.

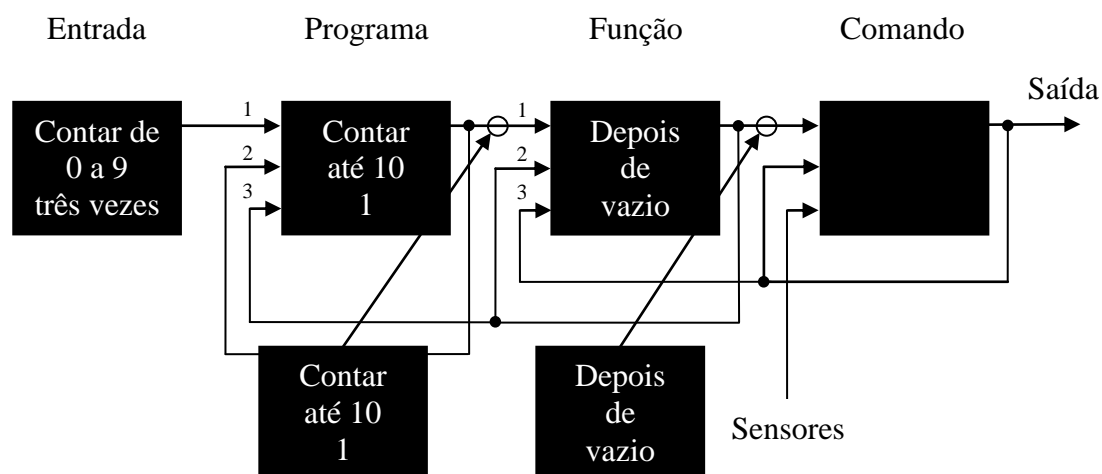


Figura 7: Passo (iv) do programa “Contar de 0 a 9 três vezes”

No passo (v) de nosso programa, Figura 8, treinamos a Camada Neural Comando para gerar a saída “0” quando observar nas suas entradas 1, 2 e 3 as imagens “Depois de vazio”, vazio e vazio, respectivamente (a entrada “Sensores” não é utilizada neste programa).

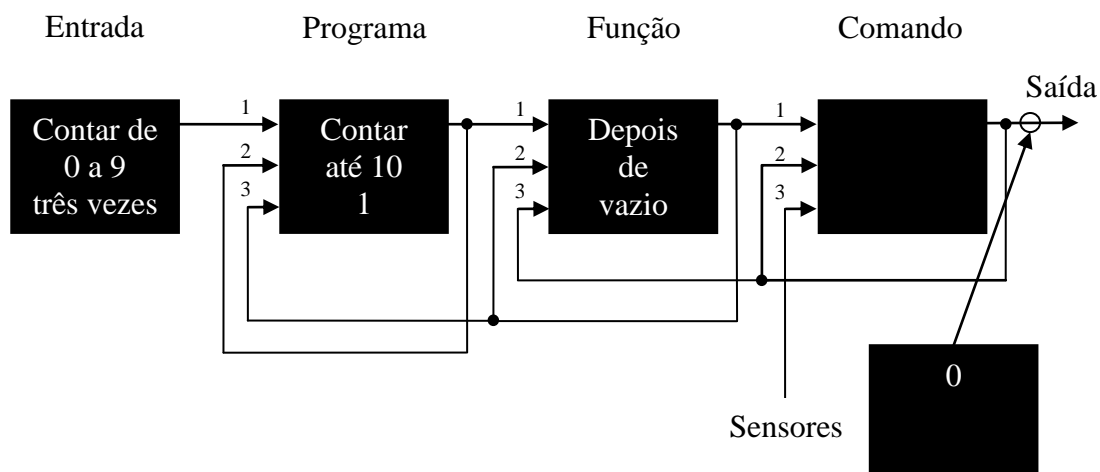


Figura 8: Passo (v) do programa “Contar de 0 a 9 três vezes”

No passo (vi) de nosso programa, Figura 9, treinamos a Camada Neural Função para gerar a saída “Depois de 0” quando observar nas suas entradas 1, 2 e 3 as imagens “Contar até 10 1”, “Depois de vazio” e “0”, respectivamente.

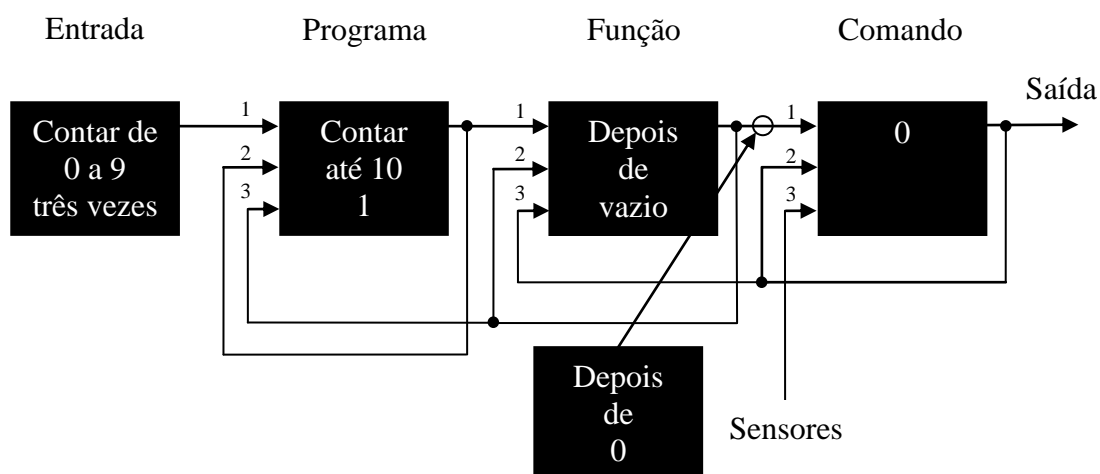


Figura 9: Passo (vi) do programa “Contar de 0 a 9 três vezes”

No passo (vii) do nosso programa, Figura 10, treinamos a Camada Neural Programa novamente para gerar como saída a imagem “Contar até 10 1” quando observar, agora, nas suas entradas 1, 2 e 3, as imagens “Contar de 0 a 9 três vezes”, “Contar até 10 1” e “Depois de 0”, respectivamente.

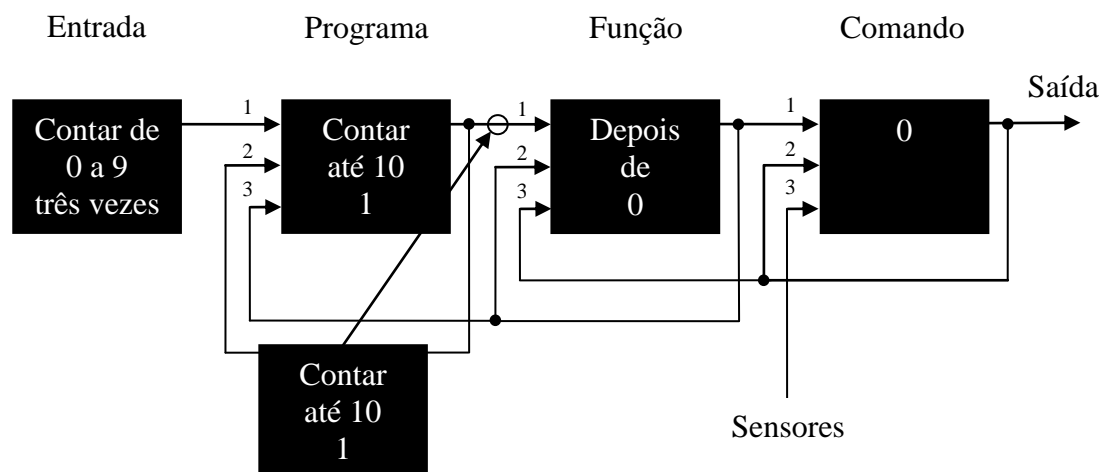


Figura 10: Passo (vii) do programa “Contar de 0 a 9 três vezes”

No passo (viii) de nosso programa, Figura 11, treinamos a Camada Neural Comando para gerar a saída “1” quando observar nas suas entradas 1, 2 e 3 as imagens “Depois de 0”, “0” e vazio, respectivamente.

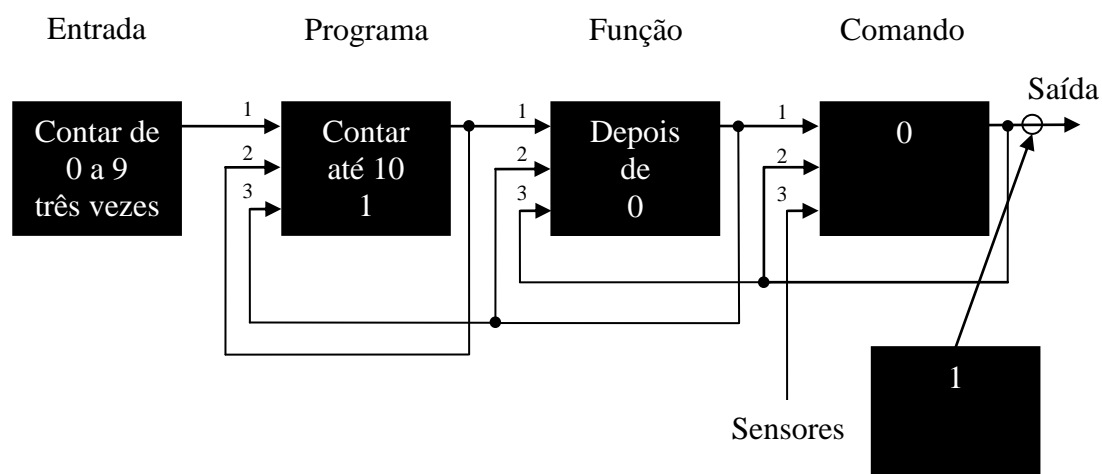


Figura 11: Passo (viii) do programa “Contar de 0 a 9 três vezes”

No passo (ix), Figura 12, treinamos a Camada Neural Função para gerar a saída “Depois de 1” quando observar nas suas entradas 1, 2 e 3 as imagens “Contar até 10 1”, “Depois de 0” e “1”, respectivamente.

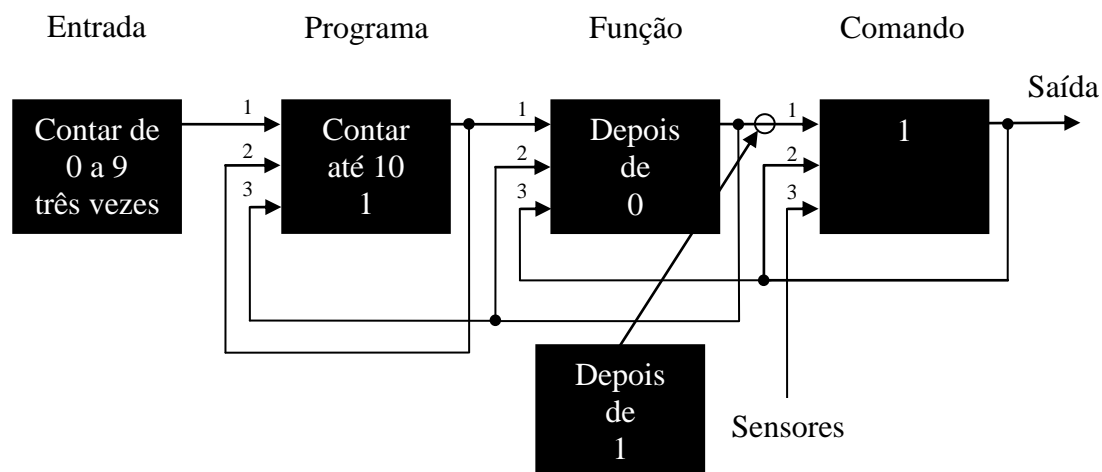


Figura 12: Passo (ix) do programa “Contar de 0 a 9 três vezes”

No passo (x) do nosso programa, Figura 13, treinamos a Camada Neural Programa novamente para gerar como saída a imagem “Contar até 10 1” quando observar, agora, nas suas entradas 1, 2 e 3, as imagens “Contar de 0 a 9 três vezes”, “Contar até 10 1” e “Depois de 1”, respectivamente.

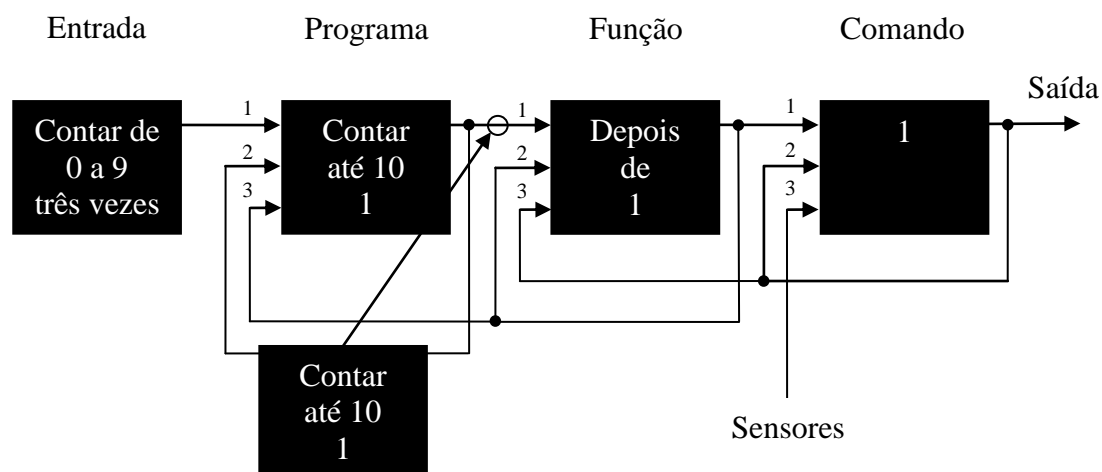


Figura 13: Passo (x) do programa “Contar de 0 a 9 três vezes”

Repetimos os passos (viii) - (x) para os números de 2 a 9 e, na situação mostrada na Figura 14, programamos o avanço da Camada Neural Programa para “Contar até 10 2”, e o recomeço das Camadas Neurais Função e Comando para uma nova contagem de 0 a 9. Procedimentos de treinamento equivalentes aos apresentados podem ser usados para avançar para “Contar até 10 3” e para finalizar o programa colocando uma imagem como “Fim de Programa” na Camada Neural Programa e vazio (quadro preto) nas demais.

Para iniciar a execução do programa basta colocar a imagem vazio (quadro preto) nas 3 Camadas Neurais e a imagem “Contar de 0 a 9 três vezes” na Entrada.

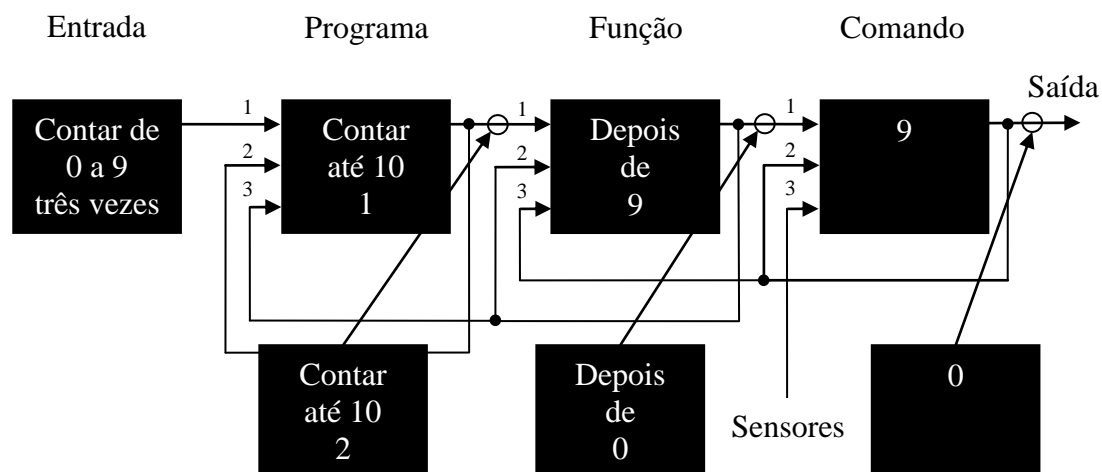


Figura 14: Passo (x) do programa “Contar de 0 a 9 três vezes”

Atenção Seletiva

Dependendo da capacidade de generalização da arquitetura da RNSP VG-RAM empregada na implementação da Camada Neural Programa, com o avanço de sua imagem de “Contar até 10 1” para “Contar até 10 2”, pode ser necessário retreinar a Camada Função para gerar corretamente cada uma das imagens (“Depois de 0” a “Depois de 9”) necessárias para cada situação possível da Camada Comando (“0” a “9”). Uma forma de evitar o risco da necessidade deste retreinamento é empregar na arquitetura do NC um mecanismo equivalente à Atenção Seletiva (pag. 135 do livro).

O mecanismo de atenção seletiva poderia ser implementado por meio da viabilização da saída das Camadas Neurais em duas modalidades (duas cores combinadas, por exemplo): uma com a função já descrita, isto é, uma imagem usada como informação de entrada de outras Camadas Neurais do NC; outra com a função de servir como máscara, que atuaria como filtro da imagem de entrada da própria Camada Neural ou de outras Camadas Neurais do NC. A segunda modalidade de saída de uma Camada Neural seria aprendida junto com primeira.

A Figura 15 ilustra o uso da máscara mencionada. Na figura, a máscara aprendida junto com cada padrão de saída aprendido pela Camada Função controla um filtro que permite apenas aos neurônios da Camada Neural Função visualizar a (ou atender à) parte “Contar até 10” da imagem “Contar até 10 1”.

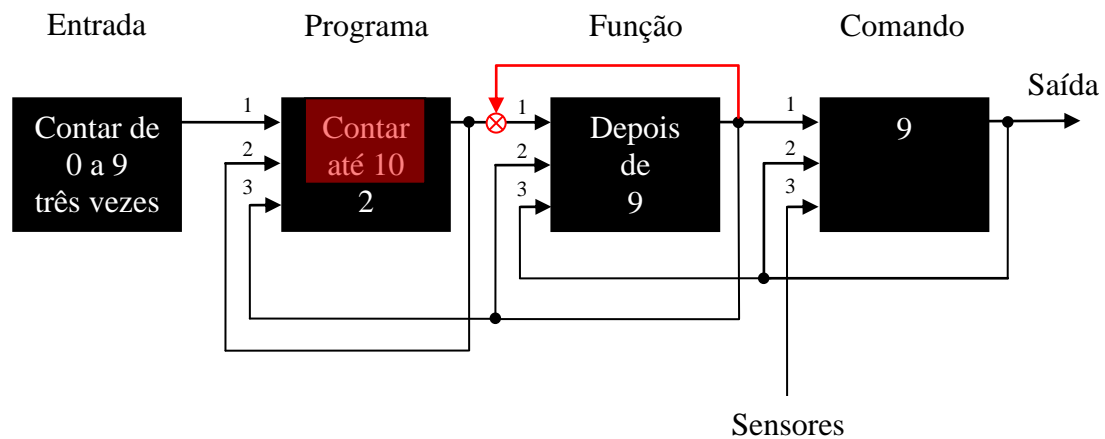


Figura 15: Mascara para viabilizar o mecanismo de Atenção Seletiva

Um filtro de atenção empregado na entrada da Camada Neural Comando como mostrado na Figura 16 pode também simplificar o treinamento desta Camada Neural. Na verdade, existem inúmeras maneiras de empregar o mecanismo de atenção para simplificar o treinamento do NC.

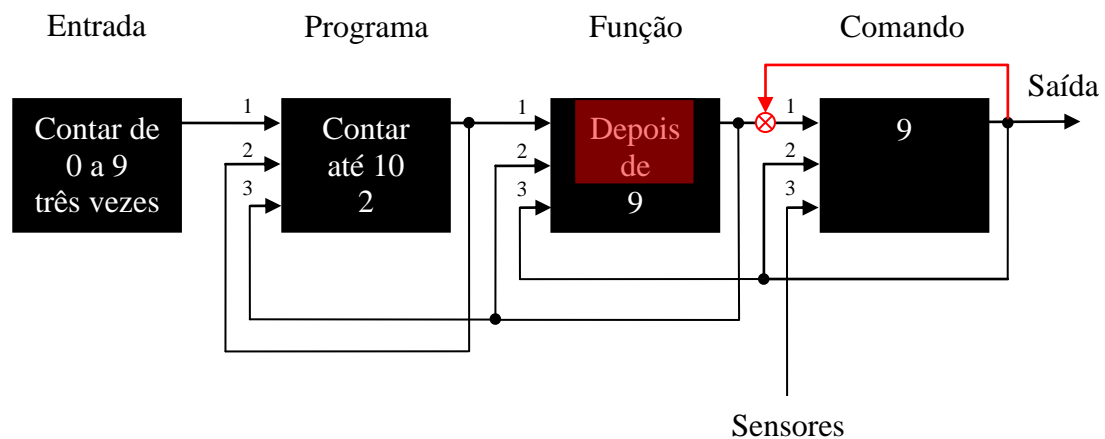


Figura 16: Outro exemplo de uso da Atenção Seletiva

Uso de um CN no Controle de um Robô Autônomo

Um CN pode ser usado para controlar um robô autônomo como mostrado na Figura 17.

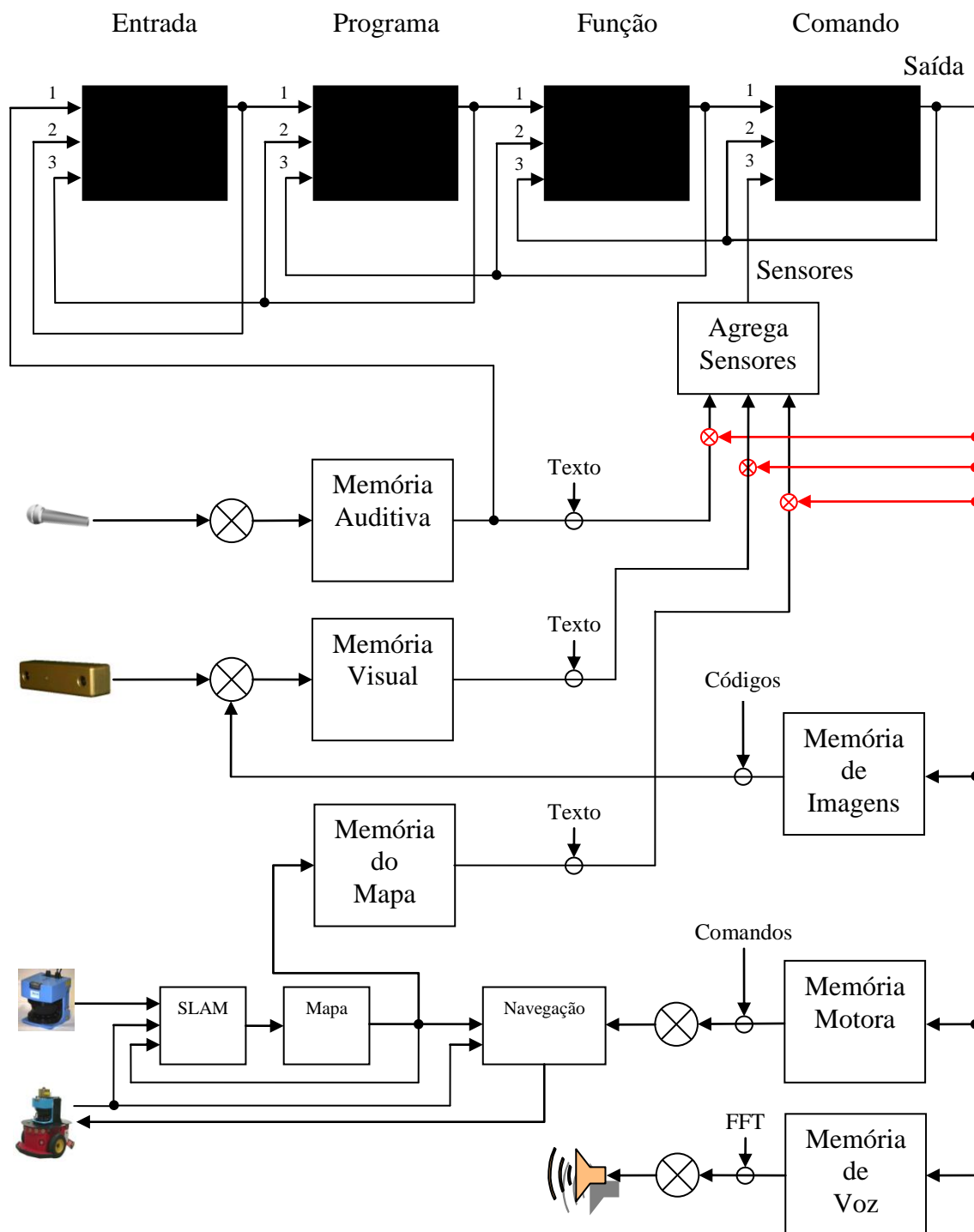


Figura 17: Arquitetura neural de controle de robôs autônomos baseada no CN

Para se comunicar com o robô o CN usa seis memórias, sendo 3 de entrada e 3 de saída. As memórias de entrada são:

- (i) **Memória Auditiva** – memória de sons captados por um microfone, pré-processados por algoritmos de interesse;
- (ii) **Memória Visual** – memória de imagens estéreo capturadas por uma câmera estéreo, pré-processadas por algoritmos de interesse;

- (iii) Memória de Mapas – memória de mapas produzidos por um algoritmo de SLAM.

As memórias de saída são:

- (iv) Memória de Imagens – memória de imagens que o CN gostaria de encontrar no campo visual das câmeras do robô;
- (v) Memória Motora – memória de comandos de navegação a serem enviados do CN para o sistema de navegação do robô;
- (vi) Memória de Voz – memória de palavras a serem ditas pelo CN.

As memórias mencionadas nos itens (i) - (vi) são implementadas com Camadas Neurais de RNSP VG-RAM como as empregadas na implementação do CN. A entrada das Camadas Neurais que implementam as memórias de entrada são imagens: (i) representando o som (tipicamente palavras) captado por um microfone; (ii) representando objetos imageados pela câmera estéreo; e (iii) representando o mapa visto de uma posição específica. As saídas aprendidas por estas Camadas Neurais são (i) imagens com textos descrevendo palavras ou outros sons captados pelo microfone, (ii) objetos imageados pela câmera estéreo, e (iii) posições no mapa. Estas saídas são agrupadas em uma única imagem pelo módulo “Agrega Sensores”. As entradas deste módulo estão sujeitas aos mecanismos de atenção previamente discutidos neste documento.

A entrada das Camadas Neurais que implementam as memórias de saída são também imagens com textos. Estas imagens, geradas pela Camada Neural Comando do CN, são coloridas, e cada memória de saída é sensível a apenas uma cor. Deste modo, a Camada Neural Comando pode enviar comandos para mais de uma memória de saída simultaneamente.

As saídas aprendidas pelas Camadas Neurais que implementam as memórias de saída são imagens: (iv) com padrões que definem um objeto específico de interesse imageado pela câmera estéreo; (v) com padrões que definem comandos de navegação, tais como “girar -10°”, “andar para frente 30cm”, ou “navegar para a posição 12, 15”; e (vi) que representam a transformada rápida de Fourier (FFT) de palavras a serem comunicadas pelo CN a humanos, respectivamente.

Loop Fonológico

O som capturado pelo microfone é filtrado (símbolo \otimes) via transformada rápida de Fourier (FFT). FFTs de sinais de voz feitas a intervalos regulares de cerca de 25ms (usualmente com sobreposição de cerca de 15ms), ou espectrogramas, podem ser agrupados ao longo do tempo de uns poucos segundos formando imagens bidimensionais. Estas imagens formam a entrada da Memória Auditiva (área de Wernicke http://en.wikipedia.org/wiki/Wernicke's_area). O treinamento da Memória Auditiva com pares espectrograma-imagem de palavras constitui um sistema de reconhecimento de voz.

A Memória Auditiva é a única ligada à Entrada do CN (ver Figura 17) e esta (a Entrada do CN) é também uma Camada Neural, que possui arquitetura de interconexões igual à dos outros níveis hierárquicos do CN. A arquitetura de interconexões da Entrada do CN permite à mesma manter uma memória do programa verbal requisitado via microfone e

equivaleria ao Loop Fonológico da mente humana (pag. 168 do livro – circuito fonológico).

É importante notar que a Memória Auditiva é também ligada à entrada Sensores do CN. Esta ligação é importante para permitir a influência de comandos de voz na execução de programas durante a execução dos mesmos.

Caminho Visual “O Que (What)”

A Memória Visual (Figura 17) busca emular o caminho *What* existente na mente humana (http://en.wikipedia.org/wiki/Two_Streams_hypothesis). Ela é também, na verdade, um sistema de reconhecimento de padrões, que gera como saída um padrão (rótulo) aprendido previamente para cada imagem estéreo de interesse apresentada em sua entrada. Ou seja, a Memória Visual permite rotular objetos no campo visual do robô. Ela permite ainda identificar em que medida estes objetos estão mais a esquerda ou a direita do robô.

Para identificar a que distância está um objeto de interesse e em que medida este objeto de interesse está mais a esquerda ou a direita do robô, uma transformada SURF é aplicada à imagem estéreo.

As *features* SURF mais proeminentes de objetos de interesse são usadas para guiar um filtro (símbolo \otimes) que leva à entrada da Memória Visual apenas o quadro associado à estas *features* do objeto de interesse. O ângulo e a distância da região da imagem com as *features* SURF do objeto de interesse são codificadas como barras na imagem de saída da Memória Visual, que permitem ao CN saber o posicionamento do objeto reconhecido no seu campo visual.

Escolha de Objetos de Interesse (Movimentos Oculares de Sacada)

A escolha do objeto de interesse é feita pelo CN por meio de imagens de saída disponibilizadas em sua Camada Neural Comando contendo texto descrevendo o objeto de interesse, que são reconhecidas pela Memória de Imagens. A Memória de Imagens é treinada com pares imagem com texto descrevendo objetos de interesse-códigos dos objetos de interesse. Todos os neurônios da Memória de Imagens aprendem o mesmo código para um dado texto descrevendo um objeto de interesse; este código é usado como um índice para uma lista de objetos de interesse que o filtro de entrada da Memória Visual (símbolo \otimes) conhece.

Caminho Visual “Onde (Where)”

O mapa usado como entrada da Memória do Mapa (Figura 17) é um mapa do tipo grid 2D centrado no robô; isto é, o robô fica sempre no centro deste mapa e apontando para frente, e o mundo é que se move com relação ao robô. Este mapa representa o caminho *Where* existente na mente humana (http://en.wikipedia.org/wiki/Two_Streams_hypothesis).

A Memória do Mapa é, na verdade, um sistema de reconhecimento de padrões, que gera como saída um padrão (rótulo) aprendido previamente para cada imagem de interesse apresentada em sua entrada. Ou seja, a Memória do Mapa permite rotular posições do mapa.

Controle do Sistema de Navegação do Robô (Córtex Motor)

A Memória Motora é treinada com pares imagem com texto descrevendo comandos motores-códigos dos comandos motores. Todos os neurônios da Memória Motora aprendem o mesmo código para um dado texto descrevendo um comando; este código é usado como um índice para uma lista de comandos de interesse (símbolo \otimes).

Geração de Mensagens de Voz (Área de Broca)

A Memória de Voz é treinada com pares imagem com texto descrevendo palavras a serem pronunciadas pelo CN-espectrogramas e representa a Área de Broca do cérebro (http://en.wikipedia.org/wiki/Broca's_area). Os espectrogramas aprendidos pela Memória de Voz são transformados em som por um filtro que implementa a transformada inversa de Fourier (símbolo \otimes).

Descrição do Trabalho

O trabalho consiste na implementação da arquitetura neural de controle de robôs autônomos baseada no CN mostrada na Figura 17 usando a MAE. O trabalho deve ser feito segundo as etapas listadas abaixo:

1. Implementação e teste do CN usando o programa exemplo “Contar de 0 a 9 três vezes” apresentado acima.
2. Implementação da Memória do Mapa e da Memória Motora e sistemas associados. Interligação da Memória do Mapa e da Memória Motora ao software de controle do robô pioneer e avaliação do desempenho do CN na proceduralização de movimentos do robô.
3. Implementação da Memória Visual e da Memória de Imagens e sistemas associados, interligação das mesmas ao software do robô Pioneer, e avaliação do desempenho do CN na proceduralização de movimentos do robô que requeiram sensoriamiento de imagens.
4. Implementação da Memória Auditiva e da Memória de Voz e sistemas associados, e avaliação do desempenho do CN na proceduralização de movimentos do robô que requeiram comunicação via voz.

O trabalho deve ser feito em conjunto por todos os alunos da disciplina. Pode ocorrer, obviamente, divisão de tarefas. Eu ajudarei a fazer o trabalho, mas COMECEM SEM MIM. ☺

Conclusão

O treinamento do CN terá que ser feito manualmente. Contudo, o ideal seria que ele ocorresse automaticamente. Mas como? Eu conjecturo que existe um nível hierárquico acima do CN que seria total ou em parte o Hipocampo (<http://en.wikipedia.org/wiki/Hippocampus>).