

Trabalho II

Inteligência Artificial

- Descrição:
 - O trabalho consiste em fazer um codificador/decodificador convolucional usando o Algoritmo de Viterbi.
 - O programa deve ler um conjunto de bits, criar o código baseado nos bits de entrada, colocar ruído no código, enviar o código para o decodificador, decodificar o código e comparar o resultado com os bits de entrada.

1 Regras

- O trabalho deve ser feito em grupos de até 3 alunos
- O trabalho deve ser feito usando C++ ou Java no Linux
- O trabalho deve ser entregue no Moodle até 05/06
- Trabalhos muito semelhantes, onde fique evidente que houve cópia, estão sujeitos a serem considerados cola. Se isto acontecer ambos os trabalhos terão nota Zero.
- Todos os materiais entregues devem ter o nome dos componentes do grupo.
- Devem ser entregue:
 - Um arquivo que compila o programa (makefile para linux, ou job para windows)
 - Os códigos (C++ ou Java) documentados
 - Um arquivo readme com qualquer instrução relevante à compilação e/ou execução do programa
 - Uma descrição detalhada de no mínimo:
 - * Descrição geral do algoritmo
 - * Descrição dos problemas e soluções usadas.
 - * Exemplos de codificação/decodificação alcançados pelo programa.

2 Critérios de Correção

- Trabalhos que não seguirem as regras anteriores tiram Zero.
- A nota do trabalho levará em conta:
 - Execução: O trabalho tem que compilar e executar
 - Completude: O trabalho faz tudo que é pedido
 - Corretude: O trabalho faz tudo de forma correta.
 - Resultado: Resultado final

3 Descrição detalhada do Algoritmo

- O algoritmo está descrito no site <http://home.netcom.com/~chip.f/viterbi/algrthms.html#algorithms>, mas abaixo é dado um resumo do algoritmo.
- Etapas:
 - Codificação: Cada bit de entrada influencia a geração de 3 pares de bits de saída, no entanto, para cada bit de entrada são gerados 2 bits de saída. Isto ocorre porque cada bit de entrada além de gerar os bits de saída, coloca o estado atual em um de 4 estados possíveis (00,01,10,11). Além disto são gerados 2 pares de bits extras no final para usar toda a influencia dos bits de entrada. O estado inicial do codificador é 00, e ele troca de estado e emite um par de bits conforme as tabelas abaixo:

Estado Atual	Par emitido		Estado Atual	Próximo estado	
	Ent=0	Ent=1		Ent=0	Ent=1
00	00	11	00	00	10
01	11	00	01	00	10
10	10	01	10	01	11
11	01	10	11	01	11

- Ruído: Após a geração do código deve ser aplicado ruído no código de maneira a modificar alguns bits. A quantidade de ruído deve ser lida e o tipo de ruído (Uniforme, Gaussiano, etc...) vocês implementam o que vocês quiserem.

- Decodificação: Uma vez recebida a mensagem codificada com ruído, a mensagem deve ser decodificada usando o Algoritmo de Viterbi conforme descrito a seguir.

Por exemplo: dado que a mensagem recebida seja: 00 11 10 00 10 01 11 11 10 00 10 11 11 11 10 11.

Começando com o estado 00, vai-se fazendo as transições e calculando para cada caminho o erro acumulado conforme mostrado abaixo.

A partir de 00, a tabela de emissão permite que seja emitido 00 (caso a entrada seja 0) ou 11 (caso a entrada seja 1).

Além disto, a tabela de transição permite que o estado vá para 00 (caso a entrada seja 0) ou 10 (caso a entrada seja 1).

Como o primeiro par recebido foi 00, temos 2 opções: ficar no estado 00 com um erro de 0, ou ficar no estado 10 com erro de 2. O erro é calculado pelo número de bits diferentes entre o que seria emitidos pelo estado e os bits recebidos, neste caso:

Estado Inicial	Entrada	Emitidos	Recebidos	Estado Final	Erro
00	0	00	00	00	0
00	1	11	00	10	2

O próximo par recebido foi 11, agora temos 4 opções:

Estado Inicial	Entrada	Emitidos	Recebidos	Estado Final	Erro
00	0	00	11	00	0+2=2
00	1	11	11	10	0+0=0
10	0	10	11	01	2+1=3
10	1	01	11	11	2+1=3

Colocando em ordem do estado final fica:

Estado Inicial	Entrada	Emitidos	Recebidos	Estado Final	Erro
00	0	00	11	00	0+2=2
10	0	10	11	01	2+1=3
00	1	11	11	10	0+0=0
10	1	01	11	11	2+1=3

O próximo par recebido foi 11, agora temos:

Estado Inicial	Entrada	Emitidos	Recebidos	Estado Final	Erro
00	0	00	11	00	$2+2=4$
00	1	10	11	10	$2+1=3$
01	0	11	11	00	$3+0=3$
01	1	00	11	10	$3+2=5$
10	0	10	11	01	$0+1=1$
10	1	01	11	11	$0+1=1$
11	0	01	11	01	$3+1=4$
11	1	10	11	11	$3+1=4$

Colocando em ordem do estado final fica:

Estado Inicial	Entrada	Emitidos	Recebidos	Estado Final	Erro	Escolhido
00	0	00	11	00	$2+2=4$	
01	0	11	11	00	$3+0=3$	\leftarrow
10	0	10	11	01	$0+1=1$	\leftarrow
11	0	01	11	01	$3+1=4$	
00	1	11	11	10	$2+0=2$	\leftarrow
01	1	00	11	10	$3+2=5$	
10	1	01	11	11	$0+1=1$	\leftarrow
11	1	10	11	11	$3+1=4$	

Quando há mais de um caminho para um estado, escolhe-se o com menor erro ou caso eles tenham o mesmo erro escolhe-se um aleatoriamente. Continua-se o processo até o fim.

No fim se pega o caminho com o menor erro e faz-se o caminho inverso, Com isto se obtêm os estados mais prováveis e as transições que ocorreram entre os estados. Para achar a entrada original faz-se a emissão para cada estado que a transição indica e despreza-se os 2 últimos valores.

- Comparação: Compara-se os valores de entrada originais com os valores decodificados.

Qualquer dúvida entre em contato comigo. Não deixe para a última hora.

Bom Trabalho