**a)** The prior probabilities for the cheetah and grass were 0.1919 and 0.8081 respectively. These prior probabilities are the same as the last homework assignment. The prior probabilities for this homework assignment were calculated by maximum likelihood estimation.

The MLE for the prior distribution was calculated to be: , where is the prior probability, is the number of observations of the feature (i.e. cheetah or grass) and is the total number of observations.

In homework 1 the priors were calculated by dividing the number of elements in the training data for both cheetah and grass by the total number of training data elements. This is the same thing as the maximum likelihood estimate for the prior probabilities.

**b)** The class conditional densities PX|Y (x|cheetah) and PX|Y (x|grass) were computed for the 64 DCT coefficients using MLE under the gaussian assumption. The MLE for the mean was and the MLE for the variance was . The area of intersection of the class conditional distributions was calculated and the 8 with the smallest area of intersection were selected as the “best 8” and the 8 with the largest area of intersection were selected as the “worst 8”. The best 8 feature distributions can be seen in figure 1, and the worst 8 feature distributions can be seen in figure 2. Figure 3 displays all 64 feature distributions. The distribution for the cheetah is displayed in red and the distribution for the grass is displayed in green. The title of each subplot represents the coefficient index.

A close up of a map

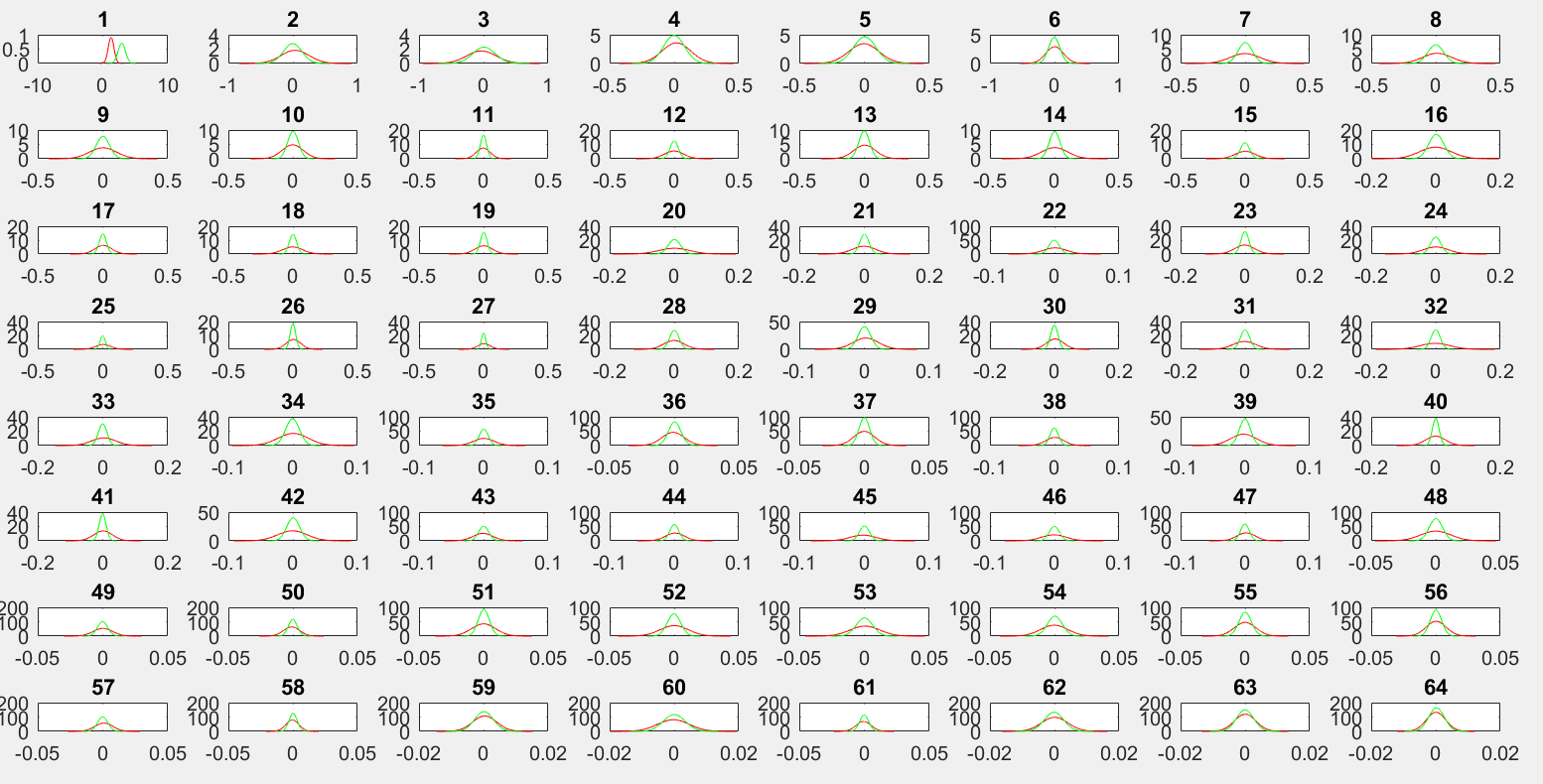
Description automatically generated

*Figure 1: Best 8 features for classification purposes*

A close up of a map

Description automatically generated

*Figure 2: Worst 8 features for classification purposes*

**

*Figure 3: All 64 feature distributions*

**c)** The Bayesian decision rule used for classification is shown in figure 4. The arguments in the first line are represented by the function g(x). The classification was done first using the 64-dimensional Gaussians and then using the best 8-dimensional Gaussians. The classification results are seen in figures 5 and 6 respectively. The probability of error for the 64-dimensional case was 8.72% and the probability of error for the 8-dimensional case was 6.48%. There were many feature distributions that had substantial overlap (e.g. the worst 8 features shown in figure 2), and the addition of these poor features do nothing but “confuse” the classifier and weaken its certainty for classification. They push the classifier to be “less strict” when choosing which pixels represent the cheetah. The “strict nature” of the 8-dimensional Gaussian classifier is represented visually by the entire neck of the cheetah being omitted. In the original image one can see that the neck area is in fact quite similar to the texture of the grass in comparison to the rest of the cheetah – too similar that the classifier didn’t “take the chance” to classify it as a part of the cheetah. However, from a visual perspective, the 8-dimensional Gaussian classifier had far fewer false positives (i.e. identifying grass as cheetah) due to its “strict nature” of classifying which ultimately resulted in a smaller error percentage. I don’t mean to anthropomorphize the classifier, but doing so aids in describing its behavior.

A close up of a logo

Description automatically generated

A close up of a logo

Description automatically generated

*Figure 4: Bayesian decision rule used for classification*

A screenshot of a cell phone

Description automatically generated

*Figure 5: 64-dimensional Gaussian classification result*

*A screenshot of a cell phone

Description automatically generated*

*Figure 6: 8-dimensional Gaussian classification result*

**Appendix**

%Joseph Bell

%ECE271A HW2

clc;

clear;

load('TrainingSamplesDCT\_8\_new.mat');

%%%%% CALCULATING PRIORS %%%%%

[rows\_FG, cols\_FG] = size(TrainsampleDCT\_FG);

[rows\_BG, cols\_BG] = size(TrainsampleDCT\_BG);

FG\_training\_elements = rows\_FG\*cols\_FG;

BG\_training\_elements = rows\_BG\*cols\_BG;

n = FG\_training\_elements + BG\_training\_elements;

% calculating prior of cheetah/foreground

z\_ik = 0;

for i=1:FG\_training\_elements

z\_ik = z\_ik + 1;

end

for i=1:BG\_training\_elements %this step is not necessary, but i'm doing it

z\_ik = z\_ik + 0; %to illustrate the maximum likelihood equation

end

prior\_cheetah = z\_ik/n; %0.1919

prior\_background = 1 - prior\_cheetah; %0.8081

%Priors are the same as last week. Reasoning explained in report.

%Calculating sample mean and variance using

%conclusion of maximum likelihood

cheetah\_zigzag = zeros(64, rows\_FG);

grass\_zigzag = zeros(64, rows\_BG);

%zigzagging each row and placing data in a column where each row is

%a DCT coefficient

for row=1:rows\_FG

cheetah\_zigzag(:,row) = zigzag(TrainsampleDCT\_FG(row,:));

end

for row=1:rows\_BG

grass\_zigzag(:,row) = zigzag(TrainsampleDCT\_BG(row,:));

end

means\_variance\_cheetah = zeros(64, 1, 2); %(:,:,1) = mean

means\_variance\_grass = zeros(64, 1, 2); %(:,:,2) = variance

%cheetah loop

for row=1:cols\_FG

N = rows\_FG;

sample\_mean = 0;

sample\_variance = 0;

sample = cheetah\_zigzag(row,:); %grab each coefficient row

for i=1:N

sample\_mean = sample\_mean + sample(1,i);

end

sample\_mean = sample\_mean/N;

for i=1:N

sample\_variance = sample\_variance + (sample(1,i) - sample\_mean)^2;

end

sample\_variance = sample\_variance/N;

means\_variance\_cheetah(row,1,1) = sample\_mean;

means\_variance\_cheetah(row,1,2) = sample\_variance;

end

%grass loop

for row=1:cols\_BG

N = rows\_BG;

sample\_mean = 0;

sample\_variance = 0;

sample = grass\_zigzag(row,:); %grab each coefficient row

for i=1:N

sample\_mean = sample\_mean + sample(1,i);

end

sample\_mean = sample\_mean/N;

for i=1:N

sample\_variance = sample\_variance + (sample(1,i) - sample\_mean)^2;

end

sample\_variance = sample\_variance/N;

means\_variance\_grass(row,1,1) = sample\_mean;

means\_variance\_grass(row,1,2) = sample\_variance;

end

overlap\_areas = zeros(1,64);

num\_of\_points = 1000;

figure(1)

for i=1:8

mean = means\_variance\_cheetah(i,1,1);

variance = means\_variance\_cheetah(i,1,2);

standard\_deviation = sqrt(variance);

values1 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation, num\_of\_points);

subplot(2,4,i)

y1 = normpdf(values1, mean, standard\_deviation);

plot(values1, y1, 'r');

hold on

mean = means\_variance\_grass(i,1,1);

variance = means\_variance\_grass(i,1,2);

standard\_deviation = sqrt(variance);

values2 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation,num\_of\_points);

y2 = normpdf(values2, mean, standard\_deviation);

plot(values2, y2, 'g');

title(i);

overlap\_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num\_of\_points);

end

figure(2)

for i=9:16

mean = means\_variance\_cheetah(i,1,1);

variance = means\_variance\_cheetah(i,1,2);

standard\_deviation = sqrt(variance);

values1 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation, num\_of\_points);

subplot(2,4,i-8)

y1 = normpdf(values1, mean, standard\_deviation);

plot(values1, y1, 'r');

hold on

mean = means\_variance\_grass(i,1,1);

variance = means\_variance\_grass(i,1,2);

standard\_deviation = sqrt(variance);

values2 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation,num\_of\_points);

y2 = normpdf(values2, mean, standard\_deviation);

plot(values2, y2, 'g');

title(i);

overlap\_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num\_of\_points);

end

figure(3)

for i=17:24

mean = means\_variance\_cheetah(i,1,1);

variance = means\_variance\_cheetah(i,1,2);

standard\_deviation = sqrt(variance);

values1 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation, num\_of\_points);

subplot(2,4,i-16)

y1 = normpdf(values1, mean, standard\_deviation);

plot(values1, y1, 'r');

hold on

mean = means\_variance\_grass(i,1,1);

variance = means\_variance\_grass(i,1,2);

standard\_deviation = sqrt(variance);

values2 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation,num\_of\_points);

y2 = normpdf(values2, mean, standard\_deviation);

plot(values2, y2, 'g');

title(i);

overlap\_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num\_of\_points);

end

figure(4)

for i=25:32

mean = means\_variance\_cheetah(i,1,1);

variance = means\_variance\_cheetah(i,1,2);

standard\_deviation = sqrt(variance);

values1 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation, num\_of\_points);

subplot(2,4,i-24)

y1 = normpdf(values1, mean, standard\_deviation);

plot(values1, y1, 'r');

hold on

mean = means\_variance\_grass(i,1,1);

variance = means\_variance\_grass(i,1,2);

standard\_deviation = sqrt(variance);

values2 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation,num\_of\_points);

y2 = normpdf(values2, mean, standard\_deviation);

plot(values2, y2, 'g');

title(i);

overlap\_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num\_of\_points);

end

figure(5)

for i=33:40

mean = means\_variance\_cheetah(i,1,1);

variance = means\_variance\_cheetah(i,1,2);

standard\_deviation = sqrt(variance);

values1 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation, num\_of\_points);

subplot(2,4,i-32)

y1 = normpdf(values1, mean, standard\_deviation);

plot(values1, y1, 'r');

hold on

mean = means\_variance\_grass(i,1,1);

variance = means\_variance\_grass(i,1,2);

standard\_deviation = sqrt(variance);

values2 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation,num\_of\_points);

y2 = normpdf(values2, mean, standard\_deviation);

plot(values2, y2, 'g');

title(i);

overlap\_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num\_of\_points);

end

figure(6)

for i=41:48

mean = means\_variance\_cheetah(i,1,1);

variance = means\_variance\_cheetah(i,1,2);

standard\_deviation = sqrt(variance);

values1 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation, num\_of\_points);

subplot(2,4,i-40)

y1 = normpdf(values1, mean, standard\_deviation);

plot(values1, y1, 'r');

hold on

mean = means\_variance\_grass(i,1,1);

variance = means\_variance\_grass(i,1,2);

standard\_deviation = sqrt(variance);

values2 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation,num\_of\_points);

y2 = normpdf(values2, mean, standard\_deviation);

plot(values2, y2, 'g');

title(i);

overlap\_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num\_of\_points);

end

figure(7)

for i=49:56

mean = means\_variance\_cheetah(i,1,1);

variance = means\_variance\_cheetah(i,1,2);

standard\_deviation = sqrt(variance);

values1 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation, num\_of\_points);

subplot(2,4,i-48)

y1 = normpdf(values1, mean, standard\_deviation);

plot(values1, y1, 'r');

hold on

mean = means\_variance\_grass(i,1,1);

variance = means\_variance\_grass(i,1,2);

standard\_deviation = sqrt(variance);

values2 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation,num\_of\_points);

y2 = normpdf(values2, mean, standard\_deviation);

plot(values2, y2, 'g');

title(i);

overlap\_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num\_of\_points);

end

figure(8)

for i=57:64

mean = means\_variance\_cheetah(i,1,1);

variance = means\_variance\_cheetah(i,1,2);

standard\_deviation = sqrt(variance);

values1 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation, num\_of\_points);

subplot(2,4,i-56)

y1 = normpdf(values1, mean, standard\_deviation);

plot(values1, y1, 'r');

hold on

mean = means\_variance\_grass(i,1,1);

variance = means\_variance\_grass(i,1,2);

standard\_deviation = sqrt(variance);

values2 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation,num\_of\_points);

y2 = normpdf(values2, mean, standard\_deviation);

plot(values2, y2, 'g');

title(i);

overlap\_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num\_of\_points);

end

figure(9)

[sorted\_overlap\_areas, indices] = sort(overlap\_areas);

top\_8 = [1 indices(1:7)];

bottom\_8 = indices(57:end);

[r, c] = size(top\_8);

for i=1:c

index = top\_8(i);

mean = means\_variance\_cheetah(index,1,1);

variance = means\_variance\_cheetah(index,1,2);

standard\_deviation = sqrt(variance);

values1 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation, num\_of\_points);

subplot(2,4,i)

y1 = normpdf(values1, mean, standard\_deviation);

plot(values1, y1, 'r');

hold on

mean = means\_variance\_grass(index,1,1);

variance = means\_variance\_grass(index,1,2);

standard\_deviation = sqrt(variance);

values2 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation,num\_of\_points);

y2 = normpdf(values2, mean, standard\_deviation);

plot(values2, y2, 'g');

title(index);

end

figure(10)

for i=1:c

index = bottom\_8(i);

mean = means\_variance\_cheetah(index,1,1);

variance = means\_variance\_cheetah(index,1,2);

standard\_deviation = sqrt(variance);

values1 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation, num\_of\_points);

subplot(2,4,i)

y1 = normpdf(values1, mean, standard\_deviation);

plot(values1, y1, 'r');

hold on

mean = means\_variance\_grass(index,1,1);

variance = means\_variance\_grass(index,1,2);

standard\_deviation = sqrt(variance);

values2 = linspace(mean-4\*standard\_deviation,mean+4\*standard\_deviation,num\_of\_points);

y2 = normpdf(values2, mean, standard\_deviation);

plot(values2, y2, 'g');

title(index);

end

mu\_cheetah = means\_variance\_cheetah(:,:,1);

mu\_grass = means\_variance\_grass(:,:,1);

x\_0 = (mu\_cheetah+mu\_grass)/2;

variance\_cheetah = means\_variance\_cheetah(:,:,2);

variance\_grass = means\_variance\_grass(:,:,2);

cheetah\_covariances = zeros(64,64);

grass\_covariances = zeros(64,64);

%calculating covariance matrix for cheetah

for i=1:64

p1 = cheetah\_zigzag(i,:); %grab coefficient row

mu\_1 = means\_variance\_cheetah(i,1,1);

for j=1:64

p2 = cheetah\_zigzag(j,:); %grab coefficient row

mu\_2 = means\_variance\_cheetah(j,1,1);

temp = 0;

for k=1:250

temp = temp + (p1(1,k) - mu\_1)\*(p2(1,k) - mu\_2);

end

cheetah\_covariances(i,j) = temp/249;

end

end

%calculating covariance matrix for grass

for i=1:64

p1 = grass\_zigzag(i,:); %grab coefficient row

mu\_1 = means\_variance\_grass(i,1,1);

for j=1:64

p2 = grass\_zigzag(j,:); %grab coefficient row

mu\_2 = means\_variance\_grass(j,1,1);

temp = 0;

for k=1:1053

temp = temp + (p1(1,k) - mu\_1)\*(p2(1,k) - mu\_2);

end

grass\_covariances(i,j) = temp/1052;

end

end

cheetah\_img = imread('cheetah.bmp');

cheetah\_img = im2double(cheetah\_img); %converting to double values since training data is of type double

[cheetah\_rows, cheetah\_cols] = size(cheetah\_img);

cheetah\_img = cheetah\_img(1:8\*floor(cheetah\_rows/8),1:8\*floor(cheetah\_cols/8)); %modifying image so it can be split into 8x8 blocks

[cheetah\_rows, cheetah\_cols] = size(cheetah\_img); %overwriting for modified dimensions

zz = load('Zig-Zag Pattern.txt');

zz = zz+1;

zz = zigzag(zz); %Credit to Alexey Sokolov from https://www.mathworks.com/matlabcentral/fileexchange/15317-zigzag-scan

%for the zig zag code

%%%%% Block Window Sliding %%%%%

new\_image64 = zeros(cheetah\_rows, cheetah\_cols);

new\_image8 = zeros(cheetah\_rows, cheetah\_cols);

for i=1:cheetah\_cols-7 %shift scan pointer over a column

for j=1:cheetah\_rows-7

block = cheetah\_img(j:7+j,i:7+i); %grab 8x8 block

block\_dct = dct2(block);

zzblock\_dct = zigzag(block\_dct);

%%%%% DO BAYESIAN DECISION RULE %%%%%

[g\_cheetah64, g\_grass64] = gaussianClassifier(transpose(zzblock\_dct),means\_variance\_cheetah,means\_variance\_grass,cheetah\_covariances,grass\_covariances,prior\_cheetah,prior\_background);

[g\_cheetah8, g\_grass8] = gaussianClassifier(transpose(zzblock\_dct(1,top\_8)),means\_variance\_cheetah(top\_8,:,:),means\_variance\_grass(top\_8,:,:),cheetah\_covariances(1:8,1:8),grass\_covariances(1:8,1:8),prior\_cheetah,prior\_background);

if g\_cheetah64 < g\_grass64

new\_image64(j:j,i:i) = 1;

end

if g\_cheetah8 < g\_grass8

new\_image8(j:j,i:i) = 1;

end

end

end

figure(12)

imagesc(new\_image64);

colormap(gray(255));

title('64 Dimensional Gaussian Result')

figure(13)

imagesc(new\_image8);

colormap(gray(255));

title('8 Dimensional Gaussian Result')

cheetah\_mask = double(imread('cheetah\_mask.bmp')/255);

counter\_correct64 = 0;

counter\_correct8 = 0;

total\_pixels = cheetah\_rows\*cheetah\_cols;

for i=1:cheetah\_rows

for j=1:cheetah\_cols

if cheetah\_mask(i,j) == new\_image64(i,j)

counter\_correct64 = counter\_correct64 + 1;

end

if cheetah\_mask(i,j) == new\_image8(i,j)

counter\_correct8 = counter\_correct8 + 1;

end

end

end

percent\_correct64 = counter\_correct64/total\_pixels\*100;

percent\_correct8 = counter\_correct8/total\_pixels\*100;

%function used to determine 8 best and 8 worst pdf combinations

function pct\_area = calculateOverlapArea(x1,x2,y1,y2,num\_points)

y\_overlap = [y1(y1<y2) y2(y2<y1)];

lesser\_x = x2(1);

greater\_x = x2(end);

if x1(1) > x2(1)

lesser\_x = x1(1);

end

if x1(end) < x2(end)

greater\_x = x1(end);

end

values = linspace(lesser\_x,greater\_x,num\_points);

area\_int = trapz(values,y\_overlap);

total\_area = trapz(x1,y1) + trapz(x2,y2);

pct\_area = area\_int/total\_area;

plot(values, y\_overlap,'b');

end

function [g\_cheetah, g\_grass] = gaussianClassifier(x,means\_variance\_cheetah,means\_variance\_grass,cheetah\_covariances,grass\_covariances,prior\_cheetah,prior\_background)

W\_I\_cheetah = inv(cheetah\_covariances);

W\_I\_grass = inv(grass\_covariances);

w\_i\_cheetah = -2\*inv(cheetah\_covariances)\*means\_variance\_cheetah(:,:,1);

w\_i\_grass = -2\*inv(grass\_covariances)\*means\_variance\_grass(:,:,1);

w\_i\_0\_cheetah = transpose(means\_variance\_cheetah(:,:,1))\*inv(cheetah\_covariances)\*means\_variance\_cheetah(:,:,1) + log(det(cheetah\_covariances))-2\*log(prior\_cheetah);

w\_i\_0\_grass = transpose(means\_variance\_grass(:,:,1))\*inv(grass\_covariances)\*means\_variance\_grass(:,:,1) + log(det(grass\_covariances))-2\*log(prior\_background);

g\_cheetah = transpose(x)\*W\_I\_cheetah\*x + transpose(w\_i\_cheetah)\*x + w\_i\_0\_cheetah;

g\_grass = transpose(x)\*W\_I\_grass\*x + transpose(w\_i\_grass)\*x + w\_i\_0\_grass;

end