Exercises 2.8 A non-convex function whose only stationary point is a global minimum

a) $g(w) = w\tanh(w)$

The code here is:

```
w=-5:0.01:5;

y=w.*tanh(w)

plot(w,y)
```

The first derivative of $g(w)$ is:

$$g'(w) = \tanh(w) + w(1 - \tanh^2(w))$$

The code here is:

```
w=-5:0.01:5;

y=tanh(w))+w.*(1-tanh(w).^2)

plot(w,y)
```

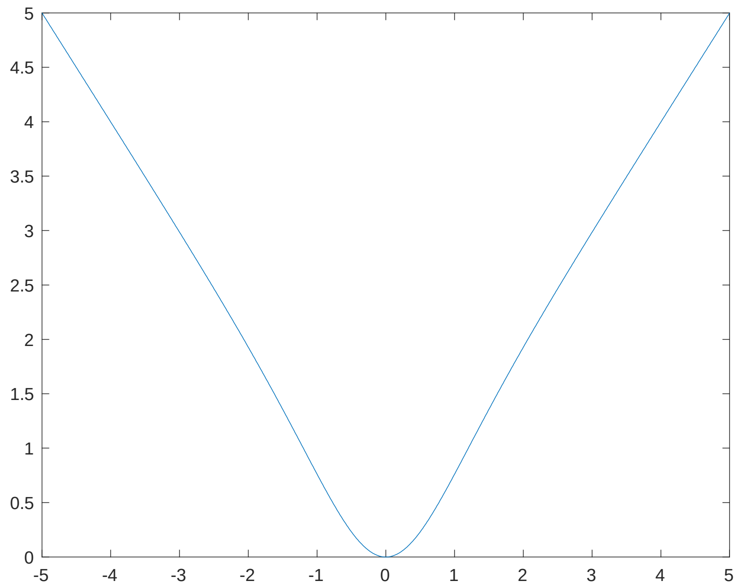

Fig.1 Graph of $g(w)$

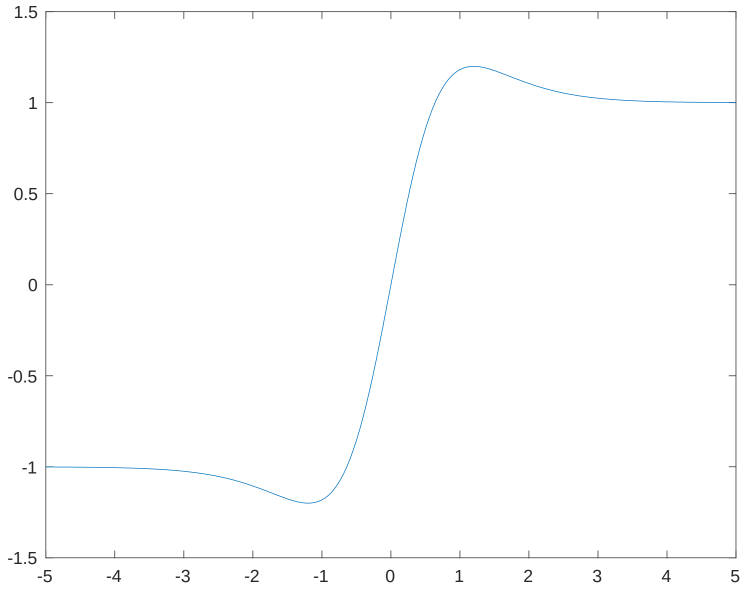

Fig.2 Graph of $g'(w)$

From Fig.2, we can see $w=0$ is the stationary point, and from Fig.1, we can see the stationary point is the global minimum of the function.

b) We can get the second derivative of $g(w)$:

$$g''(w) = 2(1-\tanh^2(w))(1-w\tanh(w))$$

The code here is:

```
w=-5:0.01:5;

y=2*(1-w.*tanh(w)).*(1-tanh(w).^2)

plot(w,y)
```

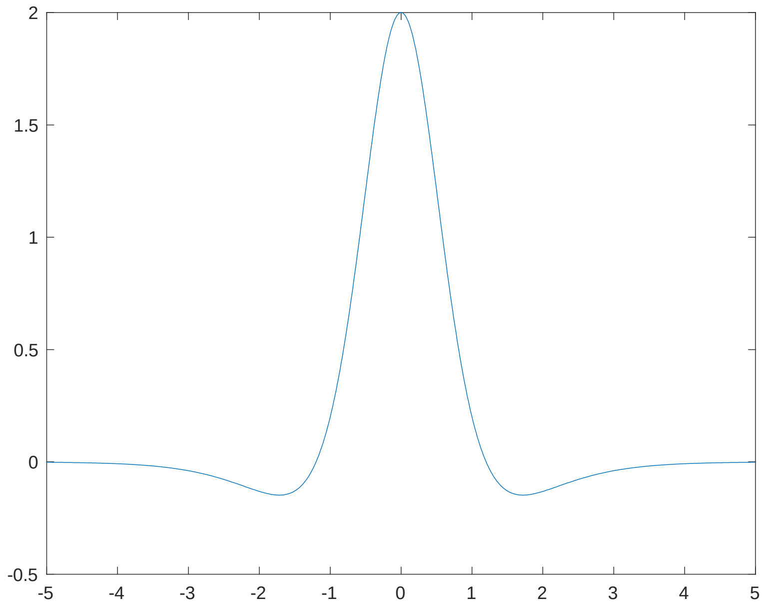

Fig.3 Graph of $g''(w)$

From Fig.3, we can see $g''(w)$ is not always greater than 0, so it is non-convex.
`

Exercises 2.13 Code up gradient descent

I used matlab to code up gradient descent, the code I added in the two_d_grad_wrapper_hw.m is:

```
grad =4*pi*w*sin(2*pi*(w'*w))+4*w;   %%% PLACE GRADIENT HERE
```

Thus, the result of gradient descent should be:

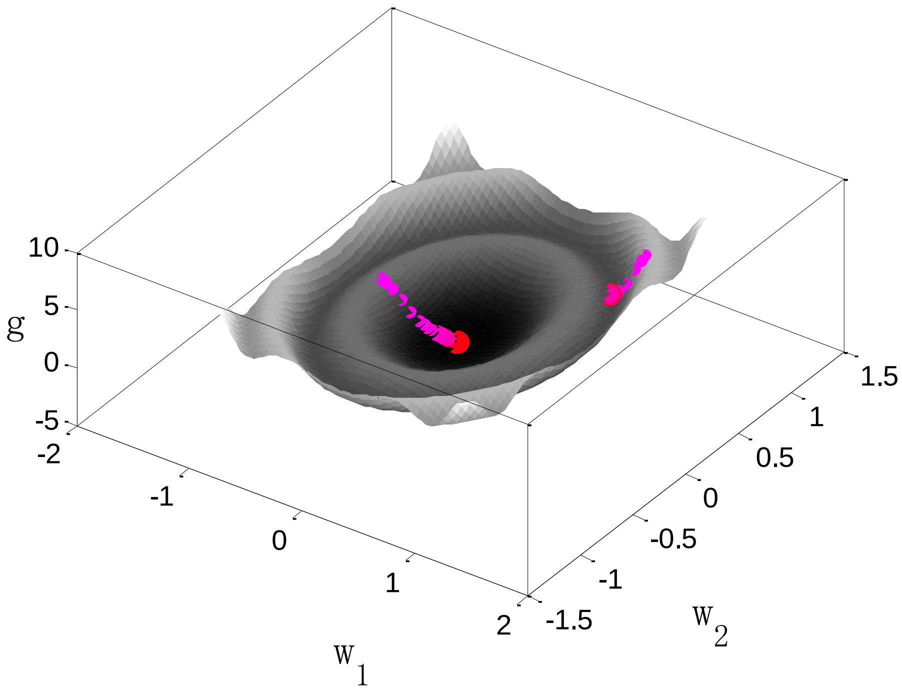

Fig4. Gradient descent

Excercises 2.17 Code up Newton's method

a) The first order condition is:

$$\frac{\alpha g(w)}{w_i} = \frac{2w_i e^{\sum_{n=1}^{N} w_n^2}}{1 + e^{\sum_{n=1}^{N} w_n^2}}$$

So, $\nabla g(w) = \dfrac{2we^{w^T w}}{1 + e^{w^T w}} = 0$

So, $w = [0,0]^T$, which is the unique stationary point of the function.

b) The surface plot of the function $g(w)$ is:

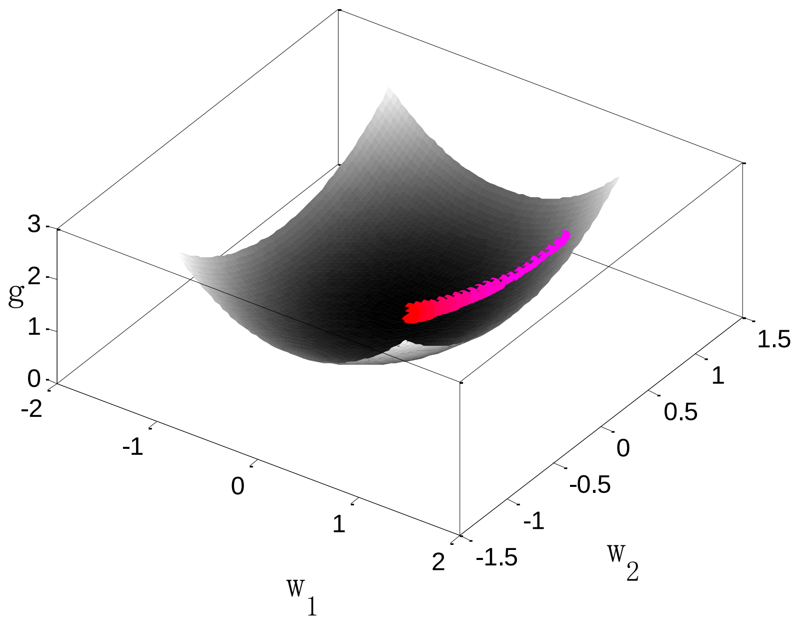

Fig.5 surface plot of $g(w)$ (Using Gradient descent)

The second derivative of $g(w)$ should be:

$$\frac{\alpha^2 g(w)}{\alpha w_i \alpha w_j} = \begin{cases} \dfrac{4w_i w_j e^{\sum_{n=1}^{N} w_n^2}}{(1+e^{\sum_{n=1}^{N} w_n^2})^2} & i \neq j \\ \dfrac{2e^{\sum_{n=1}^{N} w_n^2}(1+e^{\sum_{n=1}^{N} w_n^2}+2w_i^2)}{(1+e^{\sum_{n=1}^{N} w_n^2})^2} & i = j \end{cases}$$

So we can get the Hessian:

$$\nabla^2 g(w) = \frac{4ww^T e^{w^T w} + 2e^{w^T w}(1+e^{w^T w}) \cdot I_{N\times N}}{(1+e^{w^T w})^2}$$

Where $\mathrm{I}_{N\times N}$ is the $N\times N$ identity matrix.

So,

$$\frac{\nabla g(w)}{\nabla^2 g(w)} = \frac{\dfrac{2we^{w^T w}}{1+e^{w^T w}}}{\dfrac{4ww^T e^{w^T w} + 2e^{w^T w}(1+e^{w^T w}) \cdot \mathrm{I}_{N\times N}}{(1+e^{w^T w})^2}}$$

$$= \frac{w(1+e^{w^T w})}{2ww^T + (1+e^{w^T w}) \cdot I_{N\times N}}$$

c) $w^0 = 1_{N\times 1} = [1,1]^T$

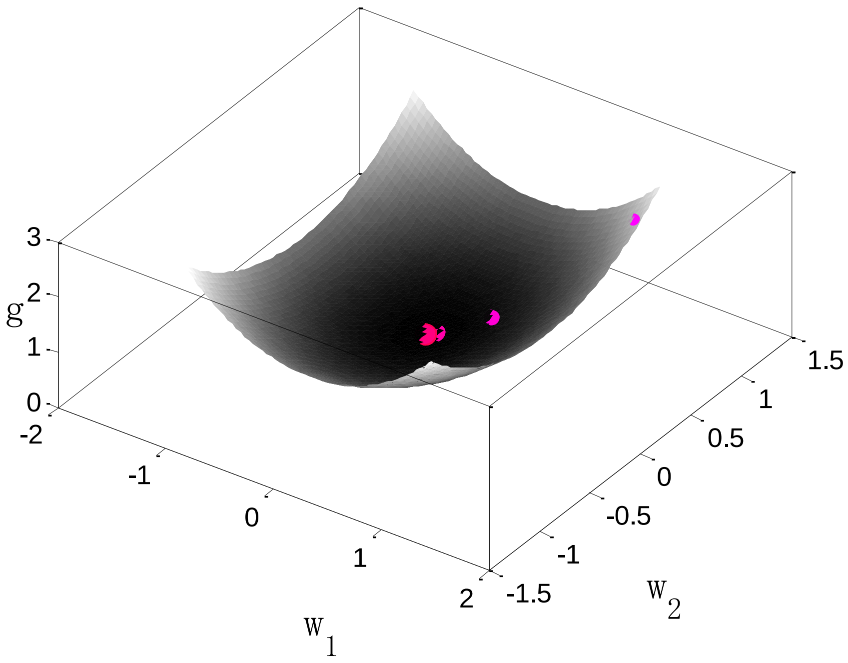

Fig. 6 Using Gradient descent

c) $w^0 = 4 \cdot 1_{N\times 1} = [4,4]^T$

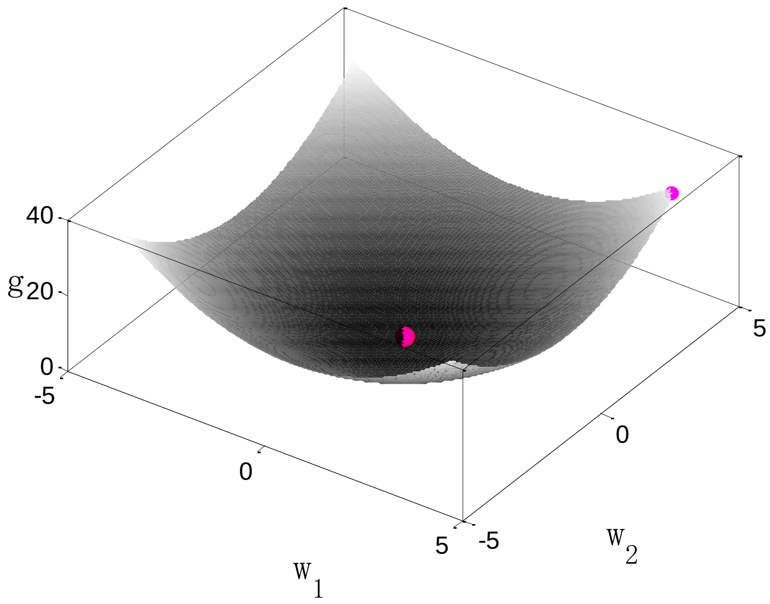

Fig. 7 Using Gradient descent

The code of c) is almost the same as d) except the initial point. So here I only show the code of d)

```
function two_d_grad_wrapper_hw()

run_all()
```

```
%%%%%%%%%%%% subfunctions %%%%%%%%%%%%
%%% performs Newton steps %%%%
   function [w,in,out] = gradient_descent(alpha,w)

   % initializations
   grad_stop = 10^-5;
   max_its = 10;
   iter = 1;
   grad = 1;
   in = [w];
    out = [log(1+exp(w'*w))];
   % main loop
   while norm(grad) > grad_stop && iter <= max_its
      grad =w'*(1+exp((w'*w)))/(2*w*w'+(1+exp((w'*w)))*[1,0;0,1]);
      w = w - grad';

      % update containers
      in = [in, w];
      out = [out, log(1+exp(w'*w))];

      % update stopers
      iter = iter + 1;
   end
end

function run_all()
   % dials for the toy
     x0 = [4;4];        % initial point (for gradient descent)
         alpha = 2*10^-3;
      %end
      %%% perform gradient descent %%%
      [x,in,out] = gradient_descent(alpha,x0);
```

```
        %%% plot function with grad descent objective evaluations %%%
        hold on
        plot_it_all(in,out)
    %end
end

%%% plots everything %%%
function plot_it_all(in,out)
    % print function
    [A,b] = make_fun();

    % print steps on surface
    plot_steps(in,out,3)
    set(gcf,'color','w');
end

%%% plots everything %%%
function [A,b] = make_fun()
    range = 4.15;                   % range over which to view surfaces
    [a1,a2] = meshgrid(-range:0.04:range);
    a1 = reshape(a1,numel(a1),1);
    a2 = reshape(a2,numel(a2),1);
    A = [a1, a2];
    A = (A.*A)*ones(2,1);
    b = log(1+exp(A))
    r = sqrt(size(b,1));
    a1 = reshape(a1,r,r);
    a2 = reshape(a2,r,r);
    b = reshape(b,r,r);
    h = surf(a1,a2,b)
    az = 35;
    el = 60;
    view(az, el);
```

```
    shading interp


    xlabel('w_1','Fontsize',18,'FontName','cmmi9')
    ylabel('w_2','Fontsize',18,'FontName','cmmi9')
    zlabel('g','Fontsize',18,'FontName','cmmi9')
    set(get(gca,'ZLabel'),'Rotation',0)
    set(gca,'FontSize',12);
    box on
    colormap gray
end

% plot descent steps on function surface
function plot_steps(in,out,dim)
    s = (1/length(out):1/length(out):1)';
    colorspec = [ones(length(out),1) ,zeros(length(out),1),flipud(s)];
    width = (1 + s)*5;
    if dim == 2
        for i = 1:length(out)
            hold on

plot(in(1,i),in(2,i),'o','Color',colorspec(i,:),'MarkerFaceColor',col
orspec(i,:),'MarkerSize',width(i));
        end
    else % dim == 3
        for i = 1:length(out)
            hold on

plot3(in(1,i),in(2,i),out(i),'o','Color',colorspec(i,:),'MarkerFaceCo
lor',colorspec(i,:),'MarkerSize',width(i));
        end
    end
end
```

```
end
```

Explanation:

$$g(w) = \log(1 + e^{w^T w}) \approx w^T w$$

When $w^T w$ is large.

So

$$h(w) = g(w^0) + \nabla g(w^0)^T (w - w^0) + \frac{1}{2}(w - w^0)^T \nabla^2 g(w^0)(w - w^0)$$
$$= w^{0T} w^0 + 2w^{0T}(w - w^0) + \frac{1}{2}(w - w^0)^T \cdot 2(w - w^0)$$
$$= w^T w$$

If $\nabla h(w)=0$, $w=[0, 0]^T$, which is a stationary point. Thus, the minimum of the second order Taylor series is the minimum of $g(w)$