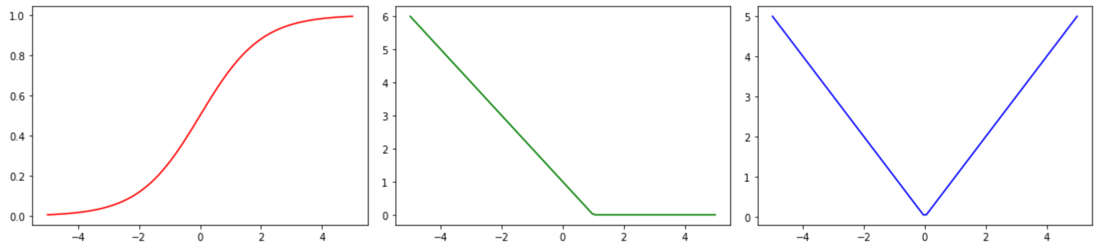
3. Gradient Calculation

Suppose **x** and y are known, $\mathbf{w} \in \mathbb{R}^d$ is a column vector. Consider the following functions that have been broadly used in machine learning.

- Sigmoid: $F(\mathbf{w}) = \frac{1}{1+e^{-\mathbf{x}\cdot\mathbf{w}}}$
- Hinge Loss: $F(\mathbf{w}) = max(0, 1 - y\mathbf{x}\cdot\mathbf{w})$
- ℓ_1-norm: $F(\mathbf{w}) = ||\mathbf{w}||_1$

1. Use python to plot their curves for the case d = 1. You can set x = y = 1.



2. Derive their gradient or subgradients for a general $d > 0$.
   To get the gradient, we differentiate the loss with respect to the i^{th} component of w.
   - Sigmoid
     - $\frac{\partial F}{\partial \mathbf{w}} = \frac{\mathbf{x}e^{-\mathbf{x}\cdot\mathbf{w}}}{(1+e^{-\mathbf{x}\cdot\mathbf{w}})^2}$
   - Hinge
     - $\frac{\partial F}{\partial \mathbf{w}} = \begin{cases} -y\,\mathbf{x} & \text{if } y\,\mathbf{x}\cdot\mathbf{w} < 1 \\ 0 & \text{if } y\,\mathbf{x}\cdot\mathbf{w} > 1 \end{cases}$
   - ℓ_1-norm
     - $\frac{\partial F}{\partial \mathbf{w}} = \frac{\mathbf{w}}{|\mathbf{w}|}$

3.1 Implementation

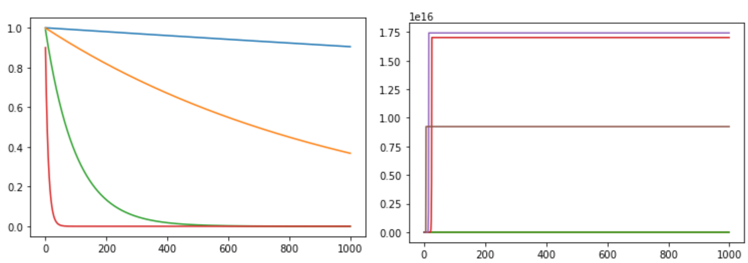
Gradient descent is typically used to solve a general optimization problem

$$min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}). \qquad (6.4)$$

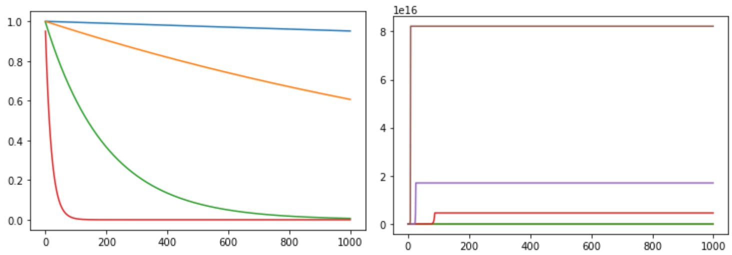
It starts from an arbitrary point $\mathbf{w}^0$ and gradually refines the solution as

$$\mathbf{w}^t \leftarrow \mathbf{w}^{t-1} - \eta \cdot \nabla F(\mathbf{w}^{t-1}).$$

Fix d = 1, i.e., the variable w is scalar. Further fix $w^0 = 1$.

1. Consider $F(w) = \frac{1}{2}w^2$. For each learning rate $\eta \in \{10^{-4}, 10^{-3}, 0.01, 0.1, 0.5, 1, 2, 5, 10, 100\}$, calculate the sequence $\{w^t\}_{t=1}^{1000}$ generated by GD and plot the curve "$|w^t|$ v.s. t".
   - 

2. Consider $F(w)=\frac{1}{4}w^2$. For each learning rate $\eta\in\{10^{-4},10^{-3},0.01,0.1,0.5,1,2,5,10,100\}$, calculate the sequence $\{w^t\}_{t=1}^{1000}$ generated by GD and plot the curve "$|w^t|$ v.s. t".

   - 

4 Linear Regression

Suppose we are given a data set $\{\mathbf{x}_i,y_i\}_{i=1}^n$ where each $\mathbf{x}_i\in\mathbb{R}^d\times\mathbb{R}$ is a row vector. We hope to learn a mapping f such that each y_i is approximated by $f(\mathbf{x}_i)$. Then a popular approach is to fit the data with *linear regression* - it assumes there exists $\mathbf{w}\in\mathbb{R}^d$ such that $y_i\approx\mathbf{w}\cdot\mathbf{x}_i$. In order to learn $\mathbf{w}$ from the data, it typically boils down to solving the following *least-squares* program:

$$min_{\mathbf{w}\in\mathbb{R}^d}F(\mathbf{w}):=\frac{1}{2}||\mathbf{y}-\mathbf{X}\mathbf{w}||^2, \qquad (4.1)$$

where $\mathbf{X}$ is the data matrix with the i^{th} row being $\mathbf{x}_i$, and $\mathbf{y}=(y_1,y_2,\dots,y_n)^\top$.

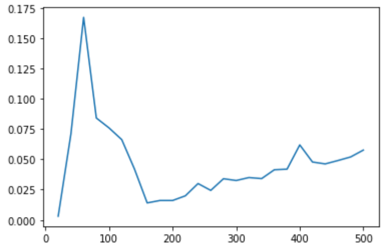
1. Compute the gradient and the Hessian matrix of $F(\mathbf{w})$, and show that (6.4) is a convex program.
   - $\frac{\partial F}{\partial\mathbf{w}}=-\mathbf{X}^\top||\mathbf{y}-\mathbf{X}\mathbf{w}||$, or $-\begin{bmatrix}\sum_{i=1}^n x_{i1}(y_i-\sum_{j=1}^d w_jx_{ij})\\ \vdots\\ \sum_{i=1}^n x_{id}(y_i-\sum_{j=1}^d w_jx_{ij})\end{bmatrix}$
   - $\nabla^2F=\frac{\partial^2F}{\partial\mathbf{w}^2}=\mathbf{X}^\top\mathbf{X}$, or $d\times n*n\times d$ $\begin{bmatrix}\sum_{i=1}^n x_{i1}^2 & \dots & \sum_{i=1}^n x_{i1}x_{id}\\ \vdots & \ddots & \vdots\\ \sum_{i=1}^n x_{id}x_{i1} & \dots & \sum_{i=1}^n x_{id}^2\end{bmatrix}$
   - $F(\mathbf{w})$ *is convex* $\iff\nabla^2F(x)\geq 0,\forall x\in D$
     - Represent a (assumedly linearly independent) $\mathbf{X}$ as $\mathbf{X}=[\mathbf{v_1}\quad\mathbf{v_2}\quad\dots\quad\mathbf{v_d}]$ $k_1\mathbf{v_1}+\dots+k_d\mathbf{v_d}=\mathbf{0}\iff k_1=\dots=k_d=0$ Now let $\mathbf{k}$ be an eigenvector of ∇^2F, $\mathbf{k}\neq\mathbf{0}\implies(k_1\mathbf{v_1}+\dots+k_d\mathbf{v_d})^2>0$
     - $=[k_1\dots k_d]\begin{bmatrix}\mathbf{v_1}\\ \vdots\\ \mathbf{v_d}\end{bmatrix}[\mathbf{v_1}\dots\mathbf{v_d}]\begin{bmatrix}k_1\\ \vdots\\ k_d\end{bmatrix}=\mathbf{k}^\top\nabla^2F\mathbf{k}=\lambda\mathbf{k}^\top\mathbf{k}>0$
     - $\lambda\mathbf{k}^\top\mathbf{k}>0\to\mathbf{k}^\top\mathbf{k}=\sum_{i=1}^d k_i^2>0\implies\lambda>0$ Since k is arbitrary, all eigenvalues must be positive, and thus ∇^2F is positive definite, which means it is also positive semi-definite.
     - Therefore, $F(\mathbf{w})$ is convex.
2. Note that (6.4) is equivalent to the following:

$$min_{\mathbf{w}\in\mathbb{R}^d}||\mathbf{y}-\mathbf{X}\mathbf{w}||^{100},$$

in the sense that any minimizer of (6.4) is also an optimum of the above, and vice versa. State why we stick with the least-squares formulation.

  - We stick with the least squares regression rather than least n^{100} regression because least squares has the lowest sampling variance while maintaining the minimum variance among all linear unbiased estimators.

3. State when the objective function is strongly-convex and when it is not.
  - A Function is strongly convex if, for any w_1, w_2, $||\nabla F(w_2) - \nabla F(w_1)||_2 \geq \alpha||w_2 - w_1||_2$, where α is the min eigenvalue of $\nabla^2 F(w)$. OLS is strongly convex if X is linearly independent.
4. Fix $n = 1000$ and increase d from 20 to 500, with a step size 20. For each problem size (n, d), generate the data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and the response $y \in \mathbb{R}^n$, for example, using the python API `numpy.random.randn`. Then calculate the exact solution $w^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$ of (6.4) and record the computation time. Plot the curve of "time v.s. d" and summarize your observation.

- 

5. Consider $n = 100$ and $d = 40$. Again, generate $\mathbf{X}$ and y, and calculate the optimal solution. Use python API to calculate the minimum and maximum eigenvalue of the Hessian matrix, and derive the upper bound on the learning rate η in gradient descent (see the slides for the bound). Let us denote this theoretical bound by η_0. Run GD on the data set with 6 choices of learning rate: $\eta \in \{0.01\eta_0, 0.1\eta_0, \eta_0, 2\eta_0, 20\eta_0, 100\eta_0\}$. Plot the curve of "$||w^t - w^*||$ v.s. t" for $1 \leq t \leq 100$ and summarize your observation. Note that you can start GD with $\mathbf{w^0} = \mathbf{0}$.
  -