Question 1: What is the computational cost of gradient descent and Newton's method? Not including the training set, how much memory is required? Suppose that computing gradients of the examples can be done in O(d) time, and express your answer in terms n and d.

Question 2: Is it guaranteed that gradient descent converges to the global optimum? Why might the convergence of gradient descent cause problems?

Question 3: What are some examples of hypotheses and loss functions that result in a convex objective?

Answer Question 1: n is the number of features, and d is the size of the dataset. For gradient descent, the computational cost per iteration is O(nd), thus the overall computational cost is O(nd*# of iterations). For Newton's method, the computational cost per iteration is O(nd^3), thus the overall computational cost is O(nd^3*# of iterations). Since Newton's method can converge faster than gradient descent, Newton's method may require fewer iterations to reach a good solution. Therefore, the overall computational cost of Newton's method is typically lower than that of gradient descent. In respect to memory, gradient descent requires storing the model parameters and takes O(d) space; Newton's method requires storing the model parameters, the Hessian matrix and its inverse, which takes O(d), O(d^2), and O(d^3) space, respectively.

Answer Question 2: No, gradient descent converges to a stationary point does not guarantee that it converges to the global optimum of the loss function. A stationary point can be either a local minimum, a local maximum, or a saddle point. For non-convex loss function, meaning that it has multiple local minima, gradient descent can get stuck in a local minimum and fail to find the global minimum.

Answer Question 3:

1. Linear regression: The hypothesis function is a linear combination of the input features, and the loss function is the mean squared error. This is a convex objective because the loss function is a quadratic function of the model parameters.
2. Logistic regression: The hypothesis function is a sigmoid function of the linear combination of the input features, and the loss function is the negative log-likelihood. This is a convex objective because the negative log-likelihood is a convex function of the model parameters.
3. Support vector machines (SVMs): The hypothesis function is a linear combination of the input features, and the loss function is the hinge loss. This is a convex objective because the hinge loss is a convex function of the model parameters.
4. Neural networks with convex activation functions: If the activation function of the neural network is a convex function, such as the rectified linear unit (ReLU), then the resulting loss function is a convex objective.
5. Regularized linear models: Adding L1 or L2 regularization to the linear regression or logistic regression objective functions also results in a convex optimization problem.