In [16]:
```
# Common imports
import numpy as np
import os
os.chdir("D:\My ML Simulations\Linear Regression")
```

In [17]:
```
# to make this notebook's output stable across runs
np.random.seed(456)
```

In [18]:
```
# To plot pretty figures
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
mpl.rc('axes', labelsize=14)
mpl.rc('xtick', labelsize=12)
mpl.rc('ytick', labelsize=12)
```

In [19]:
```
# Where to save the figures
PROJECT_ROOT_DIR = "D:\My ML Simulations\Linear Regression"
CHAPTER_ID = "training_linear_models"

def save_fig(fig_id, tight_layout=True):
    path = os.path.join(PROJECT_ROOT_DIR, "images", CHAPTER_ID, fig_id + ".png")
    print("Saving figure", fig_id)
    if tight_layout:
        plt.tight_layout()
    plt.savefig(path, format='png', dpi=300)
```

In [20]:
```
# Ignore useless warnings (see SciPy issue #5998)
import warnings
warnings.filterwarnings(action="ignore", message="^internal gelsd")
```

In [21]:
```
####   Linear regression using the Normal Equation ###

import numpy as np

X = 2 * np.random.rand(100, 1)
y = 4 + 3 * X + np.random.randn(100, 1)
```

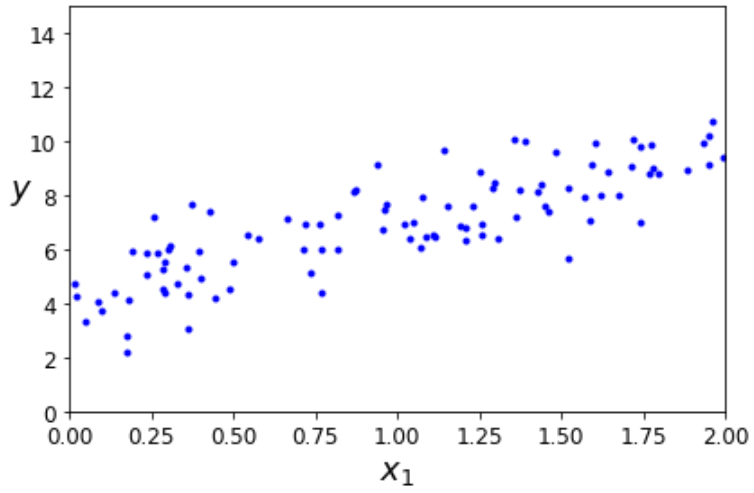
In [22]:

```
plt.plot(X, y, "b.")
plt.xlabel("$x_1$", fontsize=18)
plt.ylabel("$y$", rotation=0, fontsize=18)
plt.axis([0, 2, 0, 15])
save_fig("generated_data_plot")
plt.show()
```

```
Saving figure generated_data_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-22-e65ca261495b> in <module>
      3 plt.ylabel("$y$", rotation=0, fontsize=18)
      4 plt.axis([0, 2, 0, 15])
----> 5 save_fig("generated_data_plot")
      6 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\generated_data_plot.png'
```



In [23]:
```
X_b = np.c_[np.ones((100, 1)), X]  # add x0 = 1 to each instance
theta_best = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)
```

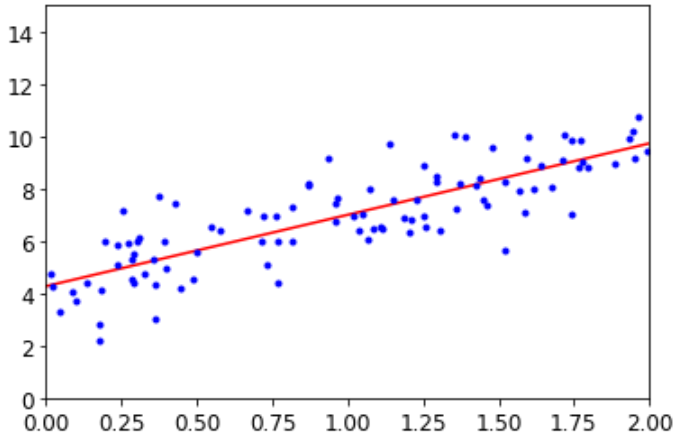
In [24]:
```
theta_best
```

Out[24]:
```
array([[4.27073388],
       [2.72876832]])
```

In [25]:
```
X_new = np.array([[0], [2]])
X_new_b = np.c_[np.ones((2, 1)), X_new]  # add x0 = 1 to each instance
y_predict = X_new_b.dot(theta_best)
y_predict
```

Out[25]:
```
array([[4.27073388],
       [9.72827051]])
```

In [26]:
```
plt.plot(X_new, y_predict, "r-")
plt.plot(X, y, "b.")
plt.axis([0, 2, 0, 15])
plt.show()
```



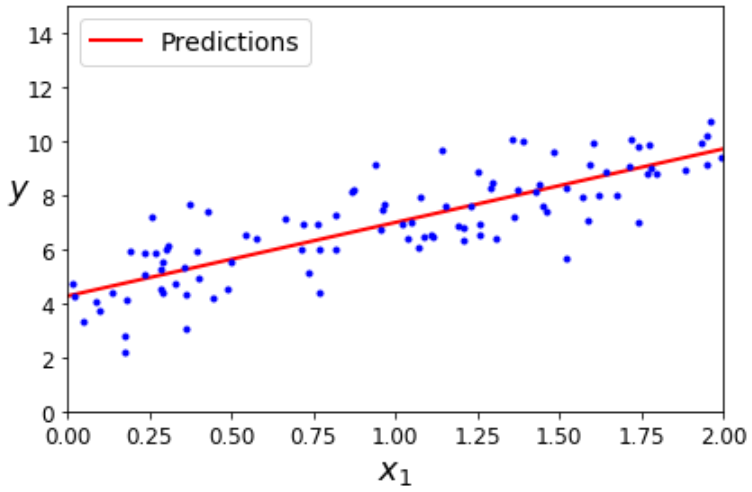
In [27]:

```
plt.plot(X_new, y_predict, "r-", linewidth=2, label="Predictions")
plt.plot(X, y, "b.")
plt.xlabel("$x_1$", fontsize=18)
plt.ylabel("$y$", rotation=0, fontsize=18)
plt.legend(loc="upper left", fontsize=14)
plt.axis([0, 2, 0, 15])
save_fig("linear_model_predictions")
plt.show()
```

```
Saving figure linear_model_predictions
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-27-c32e8491f14d> in <module>
      5 plt.legend(loc="upper left", fontsize=14)
      6 plt.axis([0, 2, 0, 15])
----> 7 save_fig("linear_model_predictions")
      8 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\linear_model_predictions.png'
```



In [29]:
```
from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(X, y)
lin_reg.intercept_, lin_reg.coef_
```

Out[29]:
```
(array([4.27073388]), array([[2.72876832]]))
```

In [30]:
```
lin_reg.predict(X_new)
```

Out[30]:
```
array([[4.27073388],
       [9.72827051]])
```

In [31]:
```
theta_best_svd, residuals, rank, s = np.linalg.lstsq(X_b, y, rcond=1e-6)
theta_best_svd
```

Out[31]:
```
array([[4.27073388],
       [2.72876832]])
```

In [32]:
```
np.linalg.pinv(X_b).dot(y)
```

Out[32]:
```
array([[4.27073388],
       [2.72876832]])
```

In [35]:
```
  ##   Linear regression using batch gradient descent

eta = 0.1
n_iterations = 1000
m = 100
theta = np.random.randn(2,1)

for iteration in range(n_iterations):
    gradients = 2/m * X_b.T.dot(X_b.dot(theta) - y)
    theta = theta - eta * gradients
```

In [36]:
```
theta
```

Out[36]:
```
array([[4.27073388],
       [2.72876832]])
```

In [37]:
```
X_new_b.dot(theta)
```

Out[37]:
```
array([[4.27073388],
       [9.72827051]])
```

In [38]:
```
theta_path_bgd = []

def plot_gradient_descent(theta, eta, theta_path=None):
    m = len(X_b)
    plt.plot(X, y, "b.")
    n_iterations = 1000
    for iteration in range(n_iterations):
        if iteration < 10:
            y_predict = X_new_b.dot(theta)
            style = "b-" if iteration > 0 else "r--"
            plt.plot(X_new, y_predict, style)
        gradients = 2/m * X_b.T.dot(X_b.dot(theta) - y)
        theta = theta - eta * gradients
        if theta_path is not None:
            theta_path.append(theta)
    plt.xlabel("$x_1$", fontsize=18)
    plt.axis([0, 2, 0, 15])
    plt.title(r"$\eta = {}$".format(eta), fontsize=16)
```

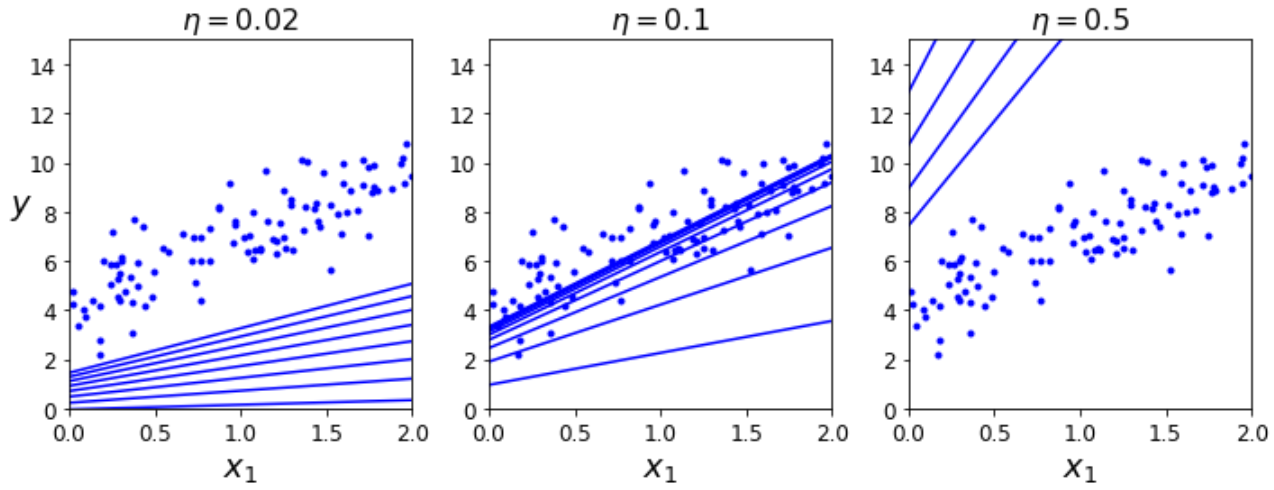
In [39]:

```
np.random.seed(456)
theta = np.random.randn(2,1)  # random initialization

plt.figure(figsize=(10,4))
plt.subplot(131); plot_gradient_descent(theta, eta=0.02)
plt.ylabel("$y$", rotation=0, fontsize=18)
plt.subplot(132); plot_gradient_descent(theta, eta=0.1, theta_path=theta_path_bgd)
plt.subplot(133); plot_gradient_descent(theta, eta=0.5)

save_fig("gradient_descent_plot")
plt.show()
```

```
Saving figure gradient_descent_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-39-34e1ef3d0335> in <module>
      8 plt.subplot(133); plot_gradient_descent(theta, eta=0.5)
      9
---> 10 save_fig("gradient_descent_plot")
     11 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\gradient_descent_plot.png'
```



In [40]:
```
###  Stochastic Gradient Descent
theta_path_sgd = []
m = len(X_b)
np.random.seed(456)
```

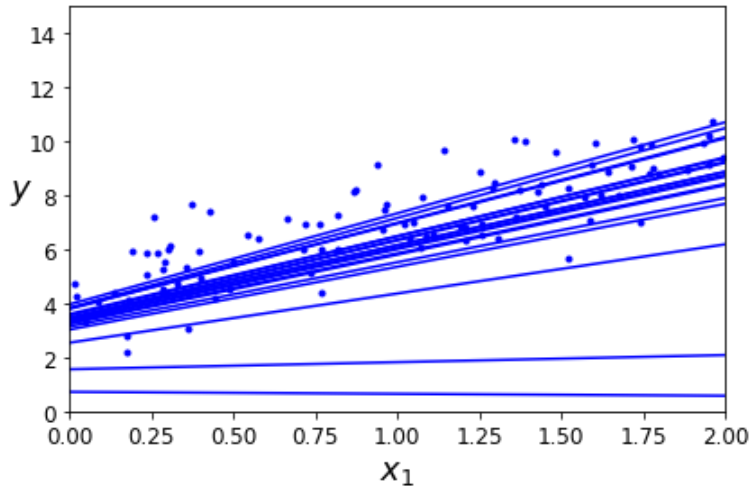
In [41]:

```
n_epochs = 50
t0, t1 = 5, 50  # learning schedule hyperparameters

def learning_schedule(t):
    return t0 / (t + t1)

theta = np.random.randn(2,1)  # random initialization

for epoch in range(n_epochs):
    for i in range(m):
        if epoch == 0 and i < 20:                    # not shown in the book
            y_predict = X_new_b.dot(theta)           # not shown
            style = "b-" if i > 0 else "r--"         # not shown
            plt.plot(X_new, y_predict, style)        # not shown
        random_index = np.random.randint(m)
        xi = X_b[random_index:random_index+1]
        yi = y[random_index:random_index+1]
        gradients = 2 * xi.T.dot(xi.dot(theta) - yi)
        eta = learning_schedule(epoch * m + i)
        theta = theta - eta * gradients
        theta_path_sgd.append(theta)                 # not shown

plt.plot(X, y, "b.")                                 # not shown
plt.xlabel("$x_1$", fontsize=18)                     # not shown
plt.ylabel("$y$", rotation=0, fontsize=18)           # not shown
plt.axis([0, 2, 0, 15])                              # not shown
save_fig("sgd_plot")                                 # not shown
plt.show()                                           # not shown
```

```
Saving figure sgd_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-41-ea71534a12f2> in <module>
     25 plt.ylabel("$y$", rotation=0, fontsize=18)           # not shown
     26 plt.axis([0, 2, 0, 15])                              # not shown
---> 27 save_fig("sgd_plot")                                 # not shown
     28 plt.show()                                           # not shown

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\sgd_plot.png'
```



In [42]:
```
theta
```

Out[42]:
```
array([[4.29656944],
       [2.7223044 ]])
```

In [43]:
```
from sklearn.linear_model import SGDRegressor
sgd_reg = SGDRegressor(max_iter=50, tol=-np.infty, penalty=None, eta0=0.1, random_state=42)
sgd_reg.fit(X, y.ravel())
```

Out[43]:
```
SGDRegressor(alpha=0.0001, average=False, early_stopping=False, epsilon=0.1,
             eta0=0.1, fit_intercept=True, l1_ratio=0.15,
             learning_rate='invscaling', loss='squared_loss', max_iter=50,
             n_iter_no_change=5, penalty=None, power_t=0.25, random_state=42,
             shuffle=True, tol=-inf, validation_fraction=0.1, verbose=0,
             warm_start=False)
```

In [44]:
```
sgd_reg.intercept_, sgd_reg.coef_
```

Out[44]:
```
(array([4.26955935]), array([2.6994705]))
```

In [45]:
```
##  Mini-batch gradient descent
theta_path_mgd = []

n_iterations = 50
minibatch_size = 20

np.random.seed(42)
theta = np.random.randn(2,1)  # random initialization

t0, t1 = 200, 1000
def learning_schedule(t):
    return t0 / (t + t1)

t = 0
for epoch in range(n_iterations):
    shuffled_indices = np.random.permutation(m)
    X_b_shuffled = X_b[shuffled_indices]
    y_shuffled = y[shuffled_indices]
    for i in range(0, m, minibatch_size):
        t += 1
        xi = X_b_shuffled[i:i+minibatch_size]
        yi = y_shuffled[i:i+minibatch_size]
        gradients = 2/minibatch_size * xi.T.dot(xi.dot(theta) - yi)
        eta = learning_schedule(t)
        theta = theta - eta * gradients
        theta_path_mgd.append(theta)
```

In [46]:
```
theta
```

Out[46]:
```
array([[4.24211315],
       [2.69049806]])
```

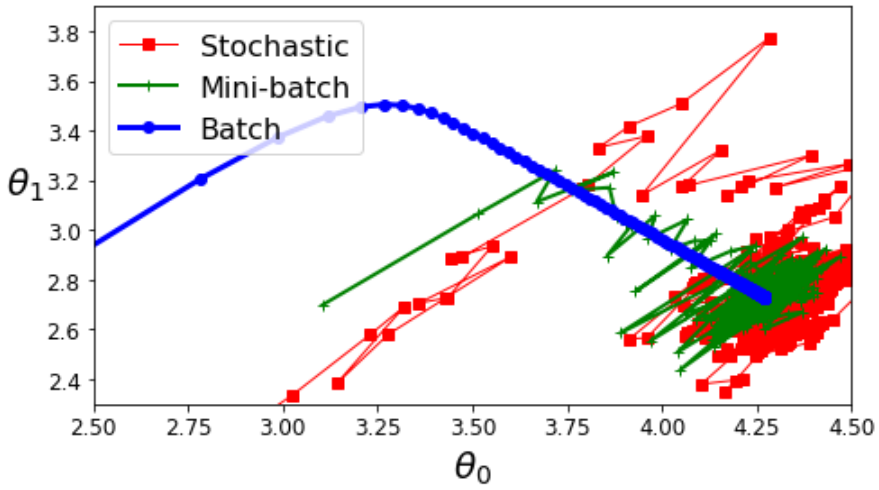
In [47]:
```
theta_path_bgd = np.array(theta_path_bgd)
theta_path_sgd = np.array(theta_path_sgd)
theta_path_mgd = np.array(theta_path_mgd)
```

In [48]:
```
plt.figure(figsize=(7,4))
plt.plot(theta_path_sgd[:, 0], theta_path_sgd[:, 1], "r-s", linewidth=1, label="Stochastic")
plt.plot(theta_path_mgd[:, 0], theta_path_mgd[:, 1], "g-+", linewidth=2, label="Mini-batch")
plt.plot(theta_path_bgd[:, 0], theta_path_bgd[:, 1], "b-o", linewidth=3, label="Batch")
plt.legend(loc="upper left", fontsize=16)
plt.xlabel(r"$\theta_0$", fontsize=20)
plt.ylabel(r"$\theta_1$    ", fontsize=20, rotation=0)
plt.axis([2.5, 4.5, 2.3, 3.9])
save_fig("gradient_descent_paths_plot")
plt.show()
```

```
Saving figure gradient_descent_paths_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-48-a403391ff799> in <module>
      7 plt.ylabel(r"$\theta_1$   ", fontsize=20, rotation=0)
      8 plt.axis([2.5, 4.5, 2.3, 3.9])
----> 9 save_fig("gradient_descent_paths_plot")
     10 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\gradient_descent_paths_plot.png'
```



In [49]:
```
 ## Polynomial regression
import numpy as np
import numpy.random as rnd

np.random.seed(456)
```

In [50]:
```
m = 100
X = 6 * np.random.rand(m, 1) - 3
y = 0.5 * X**2 + X + 2 + np.random.randn(m, 1)
```

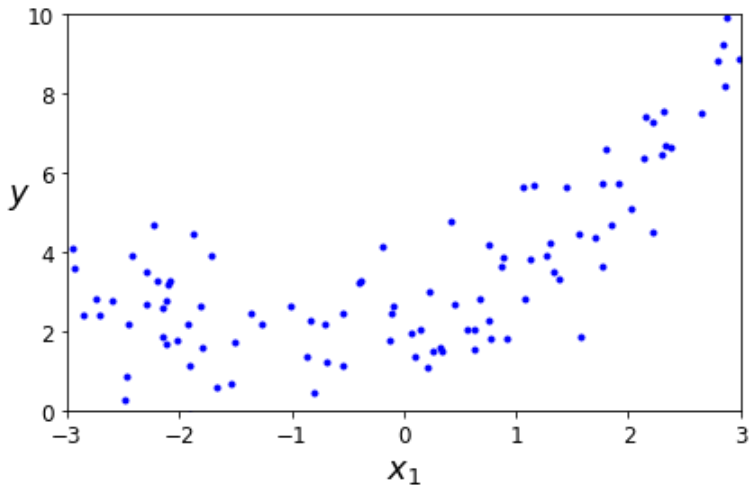
In [51]:

```
plt.plot(X, y, "b.")
plt.xlabel("$x_1$", fontsize=18)
plt.ylabel("$y$", rotation=0, fontsize=18)
plt.axis([-3, 3, 0, 10])
save_fig("quadratic_data_plot")
plt.show()
```

```
Saving figure quadratic_data_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-51-6ecc0689f32a> in <module>
      3 plt.ylabel("$y$", rotation=0, fontsize=18)
      4 plt.axis([-3, 3, 0, 10])
----> 5 save_fig("quadratic_data_plot")
      6 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\quadratic_data_plot.png'
```



In [52]:
```
from sklearn.preprocessing import PolynomialFeatures
poly_features = PolynomialFeatures(degree=2, include_bias=False)
X_poly = poly_features.fit_transform(X)
X[0]
```

Out[52]:
```
array([-1.50746454])
```

In [53]:
```
X_poly[0]
```

Out[53]:
```
array([-1.50746454,  2.27244935])
```

In [54]:
```
lin_reg = LinearRegression()
lin_reg.fit(X_poly, y)
lin_reg.intercept_, lin_reg.coef_
```

Out[54]:
```
(array([2.07432326]), array([[0.90582354, 0.47507239]]))
```

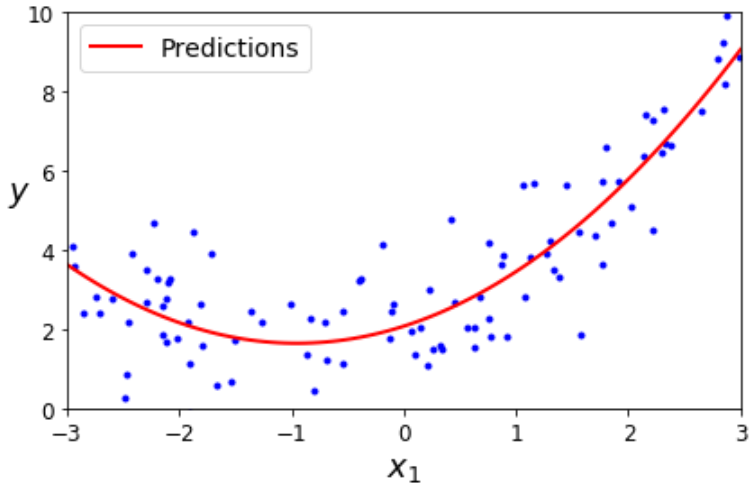
In [55]:

```
X_new=np.linspace(-3, 3, 100).reshape(100, 1)
X_new_poly = poly_features.transform(X_new)
y_new = lin_reg.predict(X_new_poly)
plt.plot(X, y, "b.")
plt.plot(X_new, y_new, "r-", linewidth=2, label="Predictions")
plt.xlabel("$x_1$", fontsize=18)
plt.ylabel("$y$", rotation=0, fontsize=18)
plt.legend(loc="upper left", fontsize=14)
plt.axis([-3, 3, 0, 10])
save_fig("quadratic_predictions_plot")
plt.show()
```

```
Saving figure quadratic_predictions_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-55-e173328d05e5> in <module>
      8 plt.legend(loc="upper left", fontsize=14)
      9 plt.axis([-3, 3, 0, 10])
---> 10 save_fig("quadratic_predictions_plot")
     11 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\quadratic_predictions_plot.png'
```



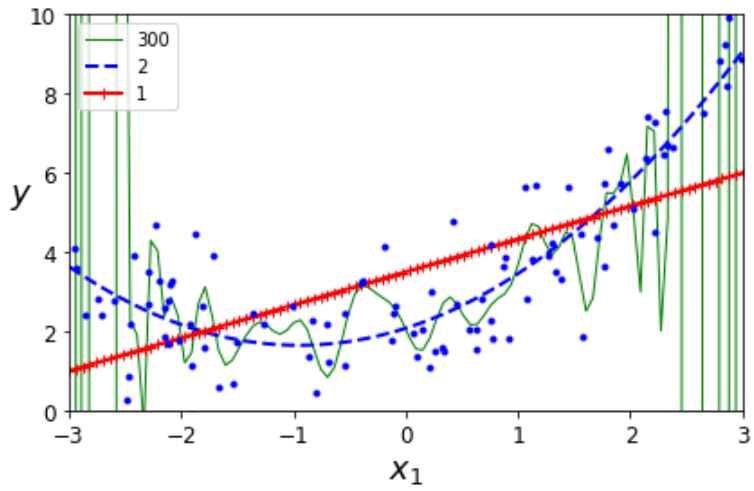
In [56]:

```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

for style, width, degree in (("g-", 1, 300), ("b--", 2, 2), ("r-+", 2, 1)):
    polybig_features = PolynomialFeatures(degree=degree, include_bias=False)
    std_scaler = StandardScaler()
    lin_reg = LinearRegression()
    polynomial_regression = Pipeline([
            ("poly_features", polybig_features),
            ("std_scaler", std_scaler),
            ("lin_reg", lin_reg),
        ])
    polynomial_regression.fit(X, y)
    y_newbig = polynomial_regression.predict(X_new)
    plt.plot(X_new, y_newbig, style, label=str(degree), linewidth=width)

plt.plot(X, y, "b.", linewidth=3)
plt.legend(loc="upper left")
plt.xlabel("$x_1$", fontsize=18)
plt.ylabel("$y$", rotation=0, fontsize=18)
plt.axis([-3, 3, 0, 10])
save_fig("high_degree_polynomials_plot")
plt.show()
```

Saving figure high_degree_polynomials_plot

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-56-b3f53474cf18> in <module>
     20 plt.ylabel("$y$", rotation=0, fontsize=18)
     21 plt.axis([-3, 3, 0, 10])
---> 22 save_fig("high_degree_polynomials_plot")
     23 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\high_degree_polynomials_plot.png'
```



In [57]:

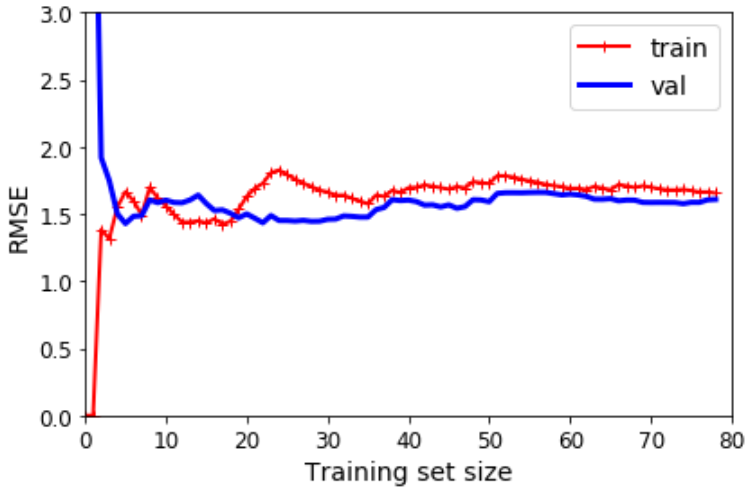
```
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split

def plot_learning_curves(model, X, y):
    X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=10)
    train_errors, val_errors = [], []
    for m in range(1, len(X_train)):
        model.fit(X_train[:m], y_train[:m])
        y_train_predict = model.predict(X_train[:m])
        y_val_predict = model.predict(X_val)
        train_errors.append(mean_squared_error(y_train[:m], y_train_predict))
        val_errors.append(mean_squared_error(y_val, y_val_predict))

    plt.plot(np.sqrt(train_errors), "r-+", linewidth=2, label="train")
    plt.plot(np.sqrt(val_errors), "b-", linewidth=3, label="val")
    plt.legend(loc="upper right", fontsize=14)   # not shown in the book
    plt.xlabel("Training set size", fontsize=14) # not shown
    plt.ylabel("RMSE", fontsize=14)              # not shown
```

In [58]:
```
lin_reg = LinearRegression()
plot_learning_curves(lin_reg, X, y)
plt.axis([0, 80, 0, 3])                         # not shown in the book
save_fig("underfitting_learning_curves_plot")   # not shown
plt.show()
```

```
Saving figure underfitting_learning_curves_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-58-57e769b88758> in <module>
      2 plot_learning_curves(lin_reg, X, y)
      3 plt.axis([0, 80, 0, 3])                         # not shown in the book
----> 4 save_fig("underfitting_learning_curves_plot")   # not shown
      5 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\underfitting_learning_curves_plot.png'
```



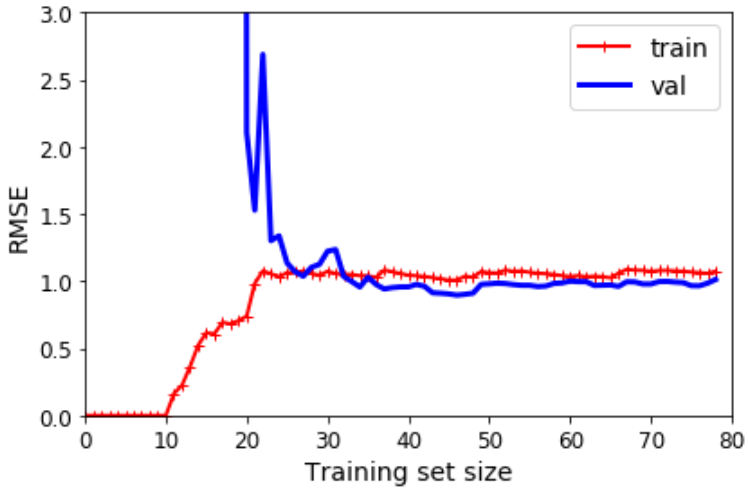
In [59]:

```
from sklearn.pipeline import Pipeline

polynomial_regression = Pipeline([
        ("poly_features", PolynomialFeatures(degree=10, include_bias=False)),
        ("lin_reg", LinearRegression()),
    ])

plot_learning_curves(polynomial_regression, X, y)
plt.axis([0, 80, 0, 3])           # not shown
save_fig("learning_curves_plot")  # not shown
plt.show()
```

```
Saving figure learning_curves_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-59-6706fba71efc> in <module>
      8 plot_learning_curves(polynomial_regression, X, y)
      9 plt.axis([0, 80, 0, 3])           # not shown
---> 10 save_fig("learning_curves_plot")  # not shown
     11 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\learning_curves_plot.png'
```



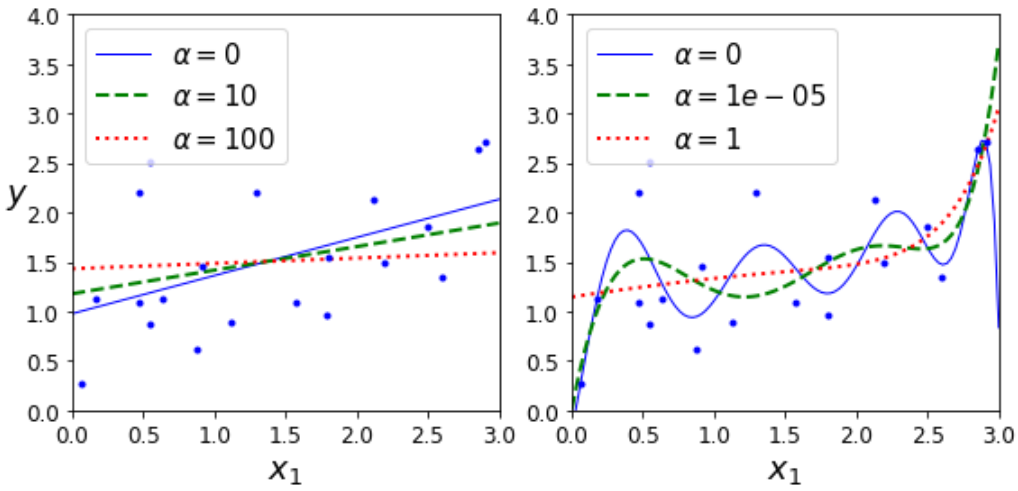
In [61]:

```
##Regularized modfrom sklearn.linear_model import Ridge

from sklearn.linear_model import Ridge

np.random.seed(42)
m = 20
X = 3 * np.random.rand(m, 1)
y = 1 + 0.5 * X + np.random.randn(m, 1) / 1.5
X_new = np.linspace(0, 3, 100).reshape(100, 1)

def plot_model(model_class, polynomial, alphas, **model_kargs):
    for alpha, style in zip(alphas, ("b-", "g--", "r:")):
        model = model_class(alpha, **model_kargs) if alpha > 0 else LinearRegression()
        if polynomial:
            model = Pipeline([
                    ("poly_features", PolynomialFeatures(degree=10, include_bias=False)),
                    ("std_scaler", StandardScaler()),
                    ("regul_reg", model),
                ])
        model.fit(X, y)
        y_new_regul = model.predict(X_new)
        lw = 2 if alpha > 0 else 1
        plt.plot(X_new, y_new_regul, style, linewidth=lw, label=r"$\alpha = {}$".format(alpha))
    plt.plot(X, y, "b.", linewidth=3)
    plt.legend(loc="upper left", fontsize=15)
    plt.xlabel("$x_1$", fontsize=18)
    plt.axis([0, 3, 0, 4])

plt.figure(figsize=(8,4))
plt.subplot(121)
plot_model(Ridge, polynomial=False, alphas=(0, 10, 100), random_state=42)
plt.ylabel("$y$", rotation=0, fontsize=18)
plt.subplot(122)
plot_model(Ridge, polynomial=True, alphas=(0, 10**-5, 1), random_state=42)

save_fig("ridge_regression_plot")
plt.show()
```

```
Saving figure ridge_regression_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-61-6f82512689ef> in <module>
     34 plot_model(Ridge, polynomial=True, alphas=(0, 10**-5, 1), random_state=42)
     35
---> 36 save_fig("ridge_regression_plot")
     37 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\ridge_regression_plot.png'
```



In [62]:
```
from sklearn.linear_model import Ridge
ridge_reg = Ridge(alpha=1, solver="cholesky", random_state=42)
ridge_reg.fit(X, y)
ridge_reg.predict([[1.5]])
```

Out[62]:
```
array([[1.55071465]])
```

In [64]:
```
sgd_reg = SGDRegressor(max_iter=50, tol=-np.infty, penalty="l2", random_state=456)
sgd_reg.fit(X, y.ravel())
sgd_reg.predict([[1.5]])
```

Out[64]:
```
array([1.49898901])
```

In [65]:
```
ridge_reg = Ridge(alpha=1, solver="sag", random_state=456)
ridge_reg.fit(X, y)
ridge_reg.predict([[1.5]])
```

Out[65]:
```
array([[1.55067453]])
```

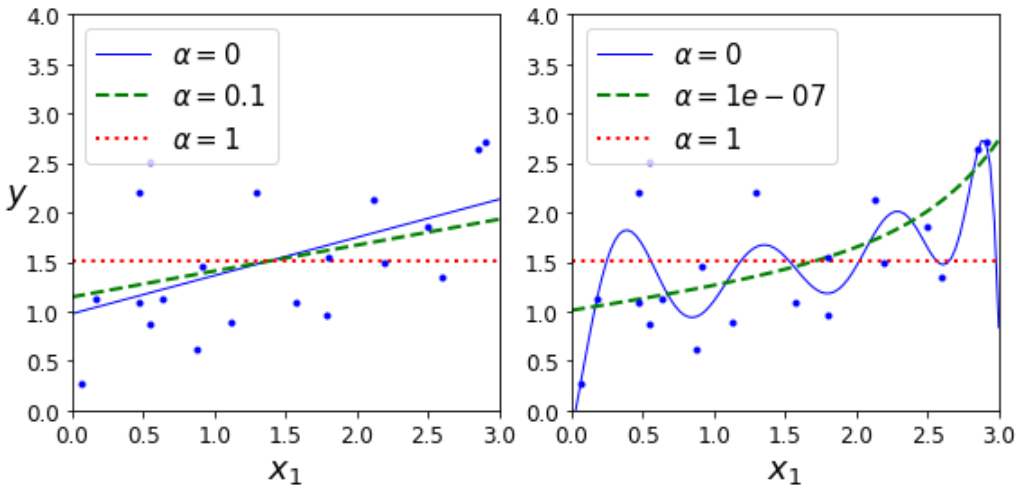
In [66]:

```
from sklearn.linear_model import Lasso

plt.figure(figsize=(8,4))
plt.subplot(121)
plot_model(Lasso, polynomial=False, alphas=(0, 0.1, 1), random_state=42)
plt.ylabel("$y$", rotation=0, fontsize=18)
plt.subplot(122)
plot_model(Lasso, polynomial=True, alphas=(0, 10**-7, 1), tol=1, random_state=42)

save_fig("lasso_regression_plot")
plt.show()
```

```
Saving figure lasso_regression_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-66-154054d3f824> in <module>
      8 plot_model(Lasso, polynomial=True, alphas=(0, 10**-7, 1), tol=1, random_state=42)
      9
---> 10 save_fig("lasso_regression_plot")
     11 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\lasso_regression_plot.png'
```



In [67]:
```
from sklearn.linear_model import Lasso
lasso_reg = Lasso(alpha=0.1)
lasso_reg.fit(X, y)
lasso_reg.predict([[1.5]])
```

Out[67]:
```
array([1.53788174])
```

In [68]:
```
from sklearn.linear_model import ElasticNet
elastic_net = ElasticNet(alpha=0.1, l1_ratio=0.5, random_state=42)
elastic_net.fit(X, y)
elastic_net.predict([[1.5]])
```

Out[68]:
```
array([1.54333232])
```

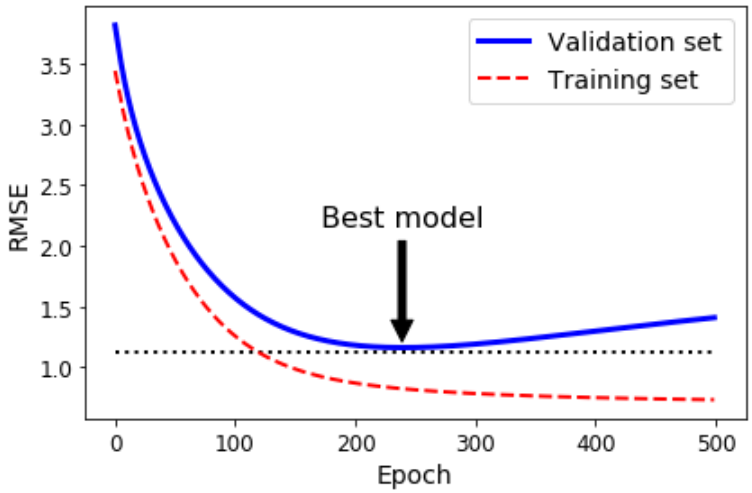
In [69]:

```
np.random.seed(42)
m = 100
X = 6 * np.random.rand(m, 1) - 3
y = 2 + X + 0.5 * X**2 + np.random.randn(m, 1)

X_train, X_val, y_train, y_val = train_test_split(X[:50], y[:50].ravel(), test_size=0.5, random_state=10)

poly_scaler = Pipeline([
        ("poly_features", PolynomialFeatures(degree=90, include_bias=False)),
        ("std_scaler", StandardScaler()),
    ])

X_train_poly_scaled = poly_scaler.fit_transform(X_train)
X_val_poly_scaled = poly_scaler.transform(X_val)

sgd_reg = SGDRegressor(max_iter=1,
                       tol=-np.infty,
                       penalty=None,
                       eta0=0.0005,
                       warm_start=True,
                       learning_rate="constant",
                       random_state=42)

n_epochs = 500
train_errors, val_errors = [], []
for epoch in range(n_epochs):
    sgd_reg.fit(X_train_poly_scaled, y_train)
    y_train_predict = sgd_reg.predict(X_train_poly_scaled)
    y_val_predict = sgd_reg.predict(X_val_poly_scaled)
    train_errors.append(mean_squared_error(y_train, y_train_predict))
    val_errors.append(mean_squared_error(y_val, y_val_predict))

best_epoch = np.argmin(val_errors)
best_val_rmse = np.sqrt(val_errors[best_epoch])

plt.annotate('Best model',
             xy=(best_epoch, best_val_rmse),
             xytext=(best_epoch, best_val_rmse + 1),
             ha="center",
             arrowprops=dict(facecolor='black', shrink=0.05),
             fontsize=16,
            )

best_val_rmse -= 0.03  # just to make the graph look better
plt.plot([0, n_epochs], [best_val_rmse, best_val_rmse], "k:", linewidth=2)
plt.plot(np.sqrt(val_errors), "b-", linewidth=3, label="Validation set")
plt.plot(np.sqrt(train_errors), "r--", linewidth=2, label="Training set")
plt.legend(loc="upper right", fontsize=14)
plt.xlabel("Epoch", fontsize=14)
plt.ylabel("RMSE", fontsize=14)
save_fig("early_stopping_plot")
plt.show()
```

```
Saving figure early_stopping_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-69-757d2e555f93> in <module>
     49 plt.xlabel("Epoch", fontsize=14)
     50 plt.ylabel("RMSE", fontsize=14)
---> 51 save_fig("early_stopping_plot")
     52 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\early_stopping_plot.png'
```



In [70]:

```
from sklearn.base import clone
sgd_reg = SGDRegressor(max_iter=1, tol=-np.infty, warm_start=True, penalty=None,
                       learning_rate="constant", eta0=0.0005, random_state=42)

minimum_val_error = float("inf")
best_epoch = None
best_model = None
for epoch in range(1000):
    sgd_reg.fit(X_train_poly_scaled, y_train)  # continues where it left off
    y_val_predict = sgd_reg.predict(X_val_poly_scaled)
    val_error = mean_squared_error(y_val, y_val_predict)
    if val_error < minimum_val_error:
        minimum_val_error = val_error
        best_epoch = epoch
        best_model = clone(sgd_reg)
```

In [71]:
```
best_epoch, best_model
```

Out[71]:
```
(239,
 SGDRegressor(alpha=0.0001, average=False, early_stopping=False, epsilon=0.1,
              eta0=0.0005, fit_intercept=True, l1_ratio=0.15,
              learning_rate='constant', loss='squared_loss', max_iter=1,
              n_iter_no_change=5, penalty=None, power_t=0.25, random_state=42,
              shuffle=True, tol=-inf, validation_fraction=0.1, verbose=0,
              warm_start=True))
```

In [72]:
```
t1a, t1b, t2a, t2b = -1, 3, -1.5, 1.5

# ignoring bias term
t1s = np.linspace(t1a, t1b, 500)
t2s = np.linspace(t2a, t2b, 500)
t1, t2 = np.meshgrid(t1s, t2s)
T = np.c_[t1.ravel(), t2.ravel()]
Xr = np.array([[-1, 1], [-0.3, -1], [1, 0.1]])
yr = 2 * Xr[:, :1] + 0.5 * Xr[:, 1:]

J = (1/len(Xr) * np.sum((T.dot(Xr.T) - yr.T)**2, axis=1)).reshape(t1.shape)

N1 = np.linalg.norm(T, ord=1, axis=1).reshape(t1.shape)
N2 = np.linalg.norm(T, ord=2, axis=1).reshape(t1.shape)

t_min_idx = np.unravel_index(np.argmin(J), J.shape)
t1_min, t2_min = t1[t_min_idx], t2[t_min_idx]

t_init = np.array([[0.25], [-1]])
```

In [73]:

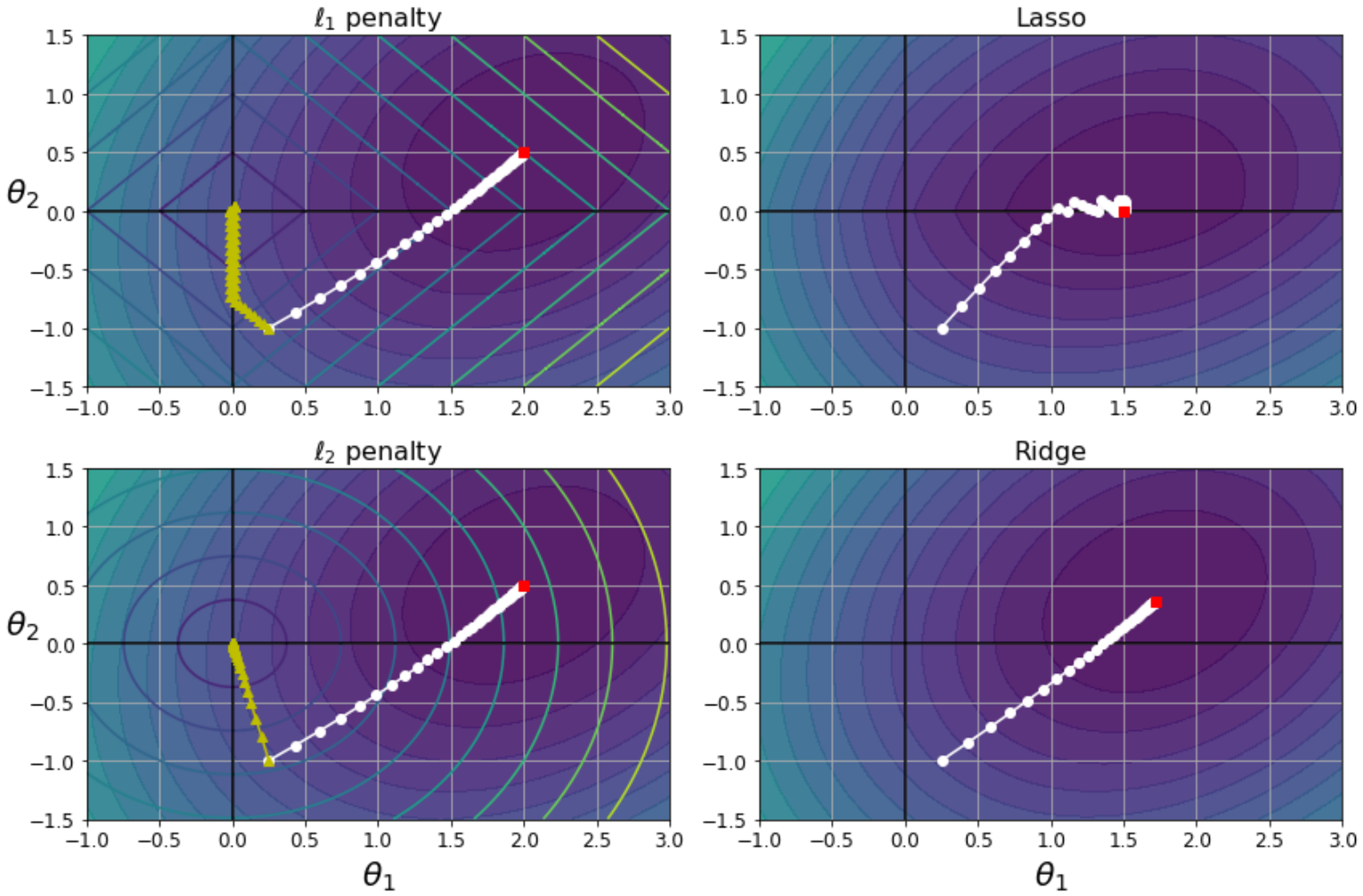
```
def bgd_path(theta, X, y, l1, l2, core = 1, eta = 0.1, n_iterations = 50):
    path = [theta]
    for iteration in range(n_iterations):
        gradients = core * 2/len(X) * X.T.dot(X.dot(theta) - y) + l1 * np.sign(theta) + 2 * l2 * theta

        theta = theta - eta * gradients
        path.append(theta)
    return np.array(path)

plt.figure(figsize=(12, 8))
for i, N, l1, l2, title in ((0, N1, 0.5, 0, "Lasso"), (1, N2, 0,  0.1, "Ridge")):
    JR = J + l1 * N1 + l2 * N2**2

    tr_min_idx = np.unravel_index(np.argmin(JR), JR.shape)
    t1r_min, t2r_min = t1[tr_min_idx], t2[tr_min_idx]

    levelsJ=(np.exp(np.linspace(0, 1, 20)) - 1) * (np.max(J) - np.min(J)) + np.min(J)
    levelsJR=(np.exp(np.linspace(0, 1, 20)) - 1) * (np.max(JR) - np.min(JR)) + np.min(JR)
    levelsN=np.linspace(0, np.max(N), 10)

    path_J = bgd_path(t_init, Xr, yr, l1=0, l2=0)
    path_JR = bgd_path(t_init, Xr, yr, l1, l2)
    path_N = bgd_path(t_init, Xr, yr, np.sign(l1)/3, np.sign(l2), core=0)

    plt.subplot(221 + i * 2)
    plt.grid(True)
    plt.axhline(y=0, color='k')
    plt.axvline(x=0, color='k')
    plt.contourf(t1, t2, J, levels=levelsJ, alpha=0.9)
    plt.contour(t1, t2, N, levels=levelsN)
    plt.plot(path_J[:, 0], path_J[:, 1], "w-o")
    plt.plot(path_N[:, 0], path_N[:, 1], "y-^")
    plt.plot(t1_min, t2_min, "rs")
    plt.title(r"$\ell_{}$ penalty".format(i + 1), fontsize=16)
    plt.axis([t1a, t1b, t2a, t2b])
    if i == 1:
        plt.xlabel(r"$\theta_1$", fontsize=20)
    plt.ylabel(r"$\theta_2$", fontsize=20, rotation=0)

    plt.subplot(222 + i * 2)
    plt.grid(True)
    plt.axhline(y=0, color='k')
    plt.axvline(x=0, color='k')
    plt.contourf(t1, t2, JR, levels=levelsJR, alpha=0.9)
    plt.plot(path_JR[:, 0], path_JR[:, 1], "w-o")
    plt.plot(t1r_min, t2r_min, "rs")
    plt.title(title, fontsize=16)
    plt.axis([t1a, t1b, t2a, t2b])
    if i == 1:
        plt.xlabel(r"$\theta_1$", fontsize=20)
```

```
save_fig("lasso_vs_ridge_plot")
plt.show()
```

```
Saving figure lasso_vs_ridge_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-73-867d0a2c7fe2> in <module>
     50         plt.xlabel(r"$\theta_1$", fontsize=20)
     51
---> 52 save_fig("lasso_vs_ridge_plot")
     53 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\lasso_vs_ridge_plot.png'
```



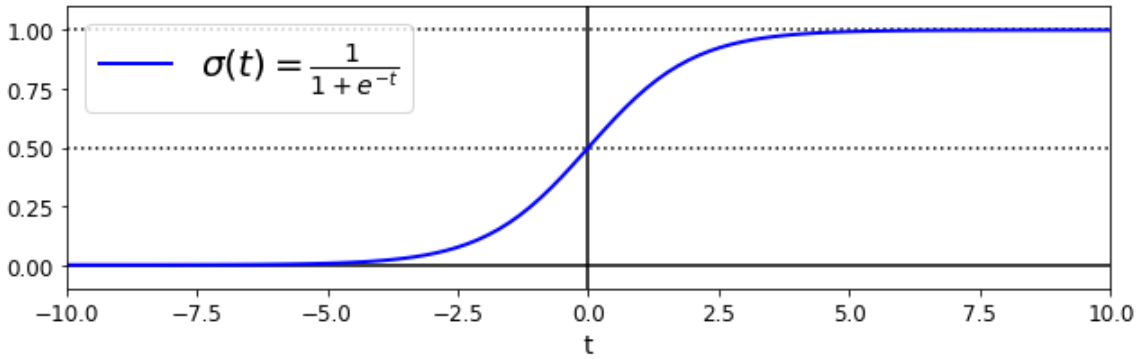
In [74]:

```
t = np.linspace(-10, 10, 100)
sig = 1 / (1 + np.exp(-t))
plt.figure(figsize=(9, 3))
plt.plot([-10, 10], [0, 0], "k-")
plt.plot([-10, 10], [0.5, 0.5], "k:")
plt.plot([-10, 10], [1, 1], "k:")
plt.plot([0, 0], [-1.1, 1.1], "k-")
plt.plot(t, sig, "b-", linewidth=2, label=r"$\sigma(t) = \frac{1}{1 + e^{-t}}$")
plt.xlabel("t")
plt.legend(loc="upper left", fontsize=20)
plt.axis([-10, 10, -0.1, 1.1])
save_fig("logistic_function_plot")
plt.show()
```

```
Saving figure logistic_function_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-74-3f9123b87e82> in <module>
     10 plt.legend(loc="upper left", fontsize=20)
     11 plt.axis([-10, 10, -0.1, 1.1])
---> 12 save_fig("logistic_function_plot")
     13 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\logistic_function_plot.png'
```



In [75]:
```
from sklearn import datasets
iris = datasets.load_iris()
list(iris.keys())
```

Out[75]:
```
['data', 'target', 'target_names', 'DESCR', 'feature_names', 'filename']
```

In [76]:
```
print(iris.DESCR)
```

.. _iris_dataset:

Iris plants dataset

Data Set Characteristics:

    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the class
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica

    :Summary Statistics:

    ============== ==== ==== ======= ===== ====================
                    Min Max   Mean   SD   Class Correlation
    ============== ==== ==== ======= ===== ====================
    sepal length: 4.3 7.9   5.84   0.83   0.7826
    sepal width: 2.0 4.4   3.05   0.43   -0.4194
    petal length: 1.0 6.9   3.76   1.76   0.9490 (high!)
    petal width: 0.1 2.5   1.20   0.76   0.9565 (high!)
    ============== ==== ==== ======= ===== ====================

    :Missing Attribute Values: None
    :Class Distribution: 33.3% for each of 3 classes.
    :Creator: R.A. Fisher
    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
    :Date: July, 1988

The famous Iris database, first used by Sir R.A. Fisher. The dataset is taken
from Fisher's paper. Note that it's the same as in R, but not as in the UCI
Machine Learning Repository, which has two wrong data points.

This is perhaps the best known database to be found in the
pattern recognition literature. Fisher's paper is a classic in the field and
is referenced frequently to this day. (See Duda & Hart, for example.) The
data set contains 3 classes of 50 instances each, where each class refers to a
type of iris plant. One class is linearly separable from the other 2; the
latter are NOT linearly separable from each other.

.. topic:: References

   - Fisher, R.A. "The use of multiple measurements in taxonomic problems"
     Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to
     Mathematical Statistics" (John Wiley, NY, 1950).

- Duda, R.O., & Hart, P.E. (1973) Pattern Classification and Scene Analysis. (Q327.D83) John Wiley & Sons. ISBN 0-471-22361-1. See page 218.
- Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System Structure and Classification Rule for Recognition in Partially Exposed Environments". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 1, 67-71.
- Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule". IEEE Transactions on Information Theory, May 1972, 431-433.
- See also: 1988 MLC Proceedings, 54-64. Cheeseman et al"s AUTOCLASS II conceptual clustering system finds 3 classes in the data.
- Many, many more ...

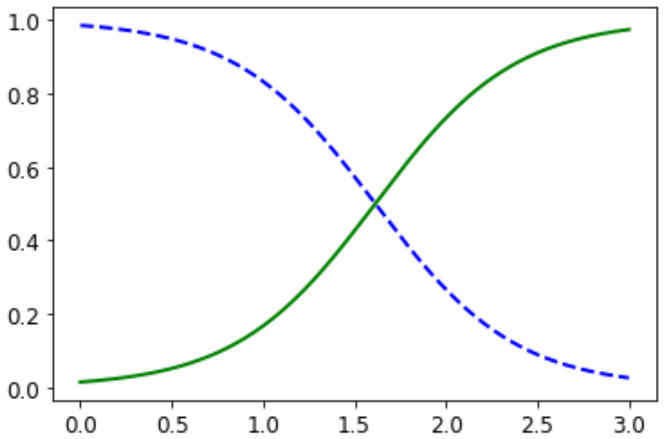
In [77]:
```
X = iris["data"][:, 3:]  # petal width
y = (iris["target"] == 2).astype(np.int)  # 1 if Iris-Virginica, else 0
```

In [78]:
```
from sklearn.linear_model import LogisticRegression
log_reg = LogisticRegression(solver="liblinear", random_state=42)
log_reg.fit(X, y)
```

Out[78]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=42, solver='liblinear', tol=0.0001, verbose=0,
                   warm_start=False)
```

In [79]:
```
X_new = np.linspace(0, 3, 1000).reshape(-1, 1)
y_proba = log_reg.predict_proba(X_new)

plt.plot(X_new, y_proba[:, 1], "g-", linewidth=2, label="Iris-Virginica")
plt.plot(X_new, y_proba[:, 0], "b--", linewidth=2, label="Not Iris-Virginica")
```

Out[79]:
```
[<matplotlib.lines.Line2D at 0xf8a14da860>]
```



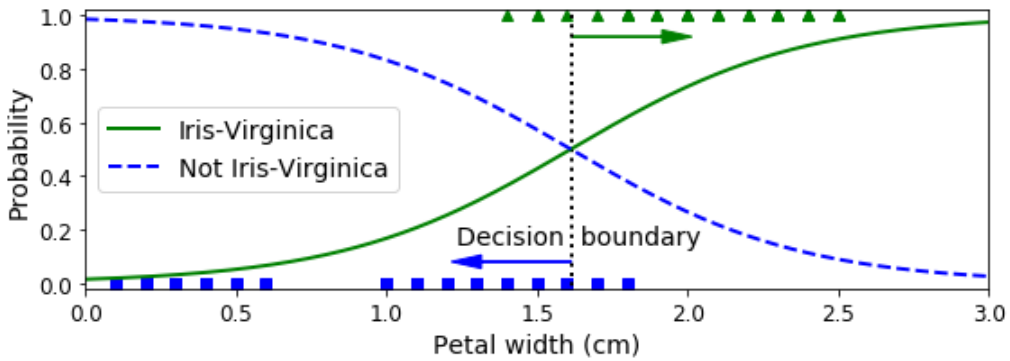
In [80]:

```
X_new = np.linspace(0, 3, 1000).reshape(-1, 1)
y_proba = log_reg.predict_proba(X_new)
decision_boundary = X_new[y_proba[:, 1] >= 0.5][0]

plt.figure(figsize=(8, 3))
plt.plot(X[y==0], y[y==0], "bs")
plt.plot(X[y==1], y[y==1], "g^")
plt.plot([decision_boundary, decision_boundary], [-1, 2], "k:", linewidth=2)
plt.plot(X_new, y_proba[:, 1], "g-", linewidth=2, label="Iris-Virginica")
plt.plot(X_new, y_proba[:, 0], "b--", linewidth=2, label="Not Iris-Virginica")
plt.text(decision_boundary+0.02, 0.15, "Decision  boundary", fontsize=14, color="k", ha="center")
plt.arrow(decision_boundary, 0.08, -0.3, 0, head_width=0.05, head_length=0.1, fc='b', ec='b')
plt.arrow(decision_boundary, 0.92, 0.3, 0, head_width=0.05, head_length=0.1, fc='g', ec='g')
plt.xlabel("Petal width (cm)", fontsize=14)
plt.ylabel("Probability", fontsize=14)
plt.legend(loc="center left", fontsize=14)
plt.axis([0, 3, -0.02, 1.02])
save_fig("logistic_regression_plot")
plt.show()
```

```
Saving figure logistic_regression_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-80-89728fd50fd9> in <module>
     16 plt.legend(loc="center left", fontsize=14)
     17 plt.axis([0, 3, -0.02, 1.02])
---> 18 save_fig("logistic_regression_plot")
     19 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\logistic_regression_plot.png'
```



In [81]:
```
decision_boundary
```

Out[81]:
```
array([1.61561562])
```

In [82]:
```
log_reg.predict([[1.7], [1.5]])
```

Out[82]:
```
array([1, 0])
```

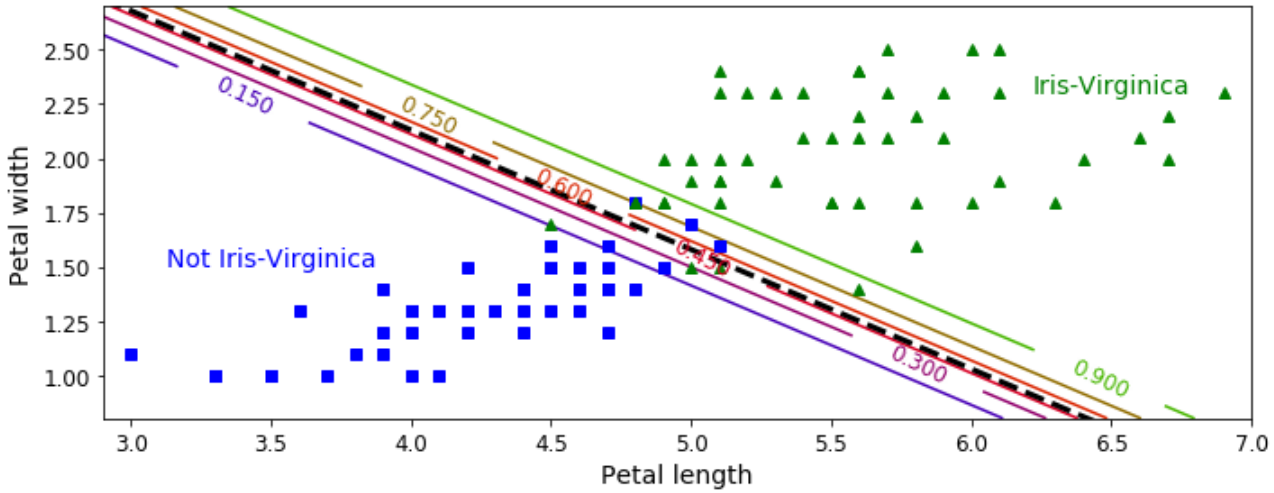
In [83]:

```
from sklearn.linear_model import LogisticRegression

X = iris["data"][:, (2, 3)]  # petal length, petal width
y = (iris["target"] == 2).astype(np.int)

log_reg = LogisticRegression(solver="liblinear", C=10**10, random_state=42)
log_reg.fit(X, y)

x0, x1 = np.meshgrid(
        np.linspace(2.9, 7, 500).reshape(-1, 1),
        np.linspace(0.8, 2.7, 200).reshape(-1, 1),
    )
X_new = np.c_[x0.ravel(), x1.ravel()]

y_proba = log_reg.predict_proba(X_new)

plt.figure(figsize=(10, 4))
plt.plot(X[y==0, 0], X[y==0, 1], "bs")
plt.plot(X[y==1, 0], X[y==1, 1], "g^")

zz = y_proba[:, 1].reshape(x0.shape)
contour = plt.contour(x0, x1, zz, cmap=plt.cm.brg)


left_right = np.array([2.9, 7])
boundary = -(log_reg.coef_[0][0] * left_right + log_reg.intercept_[0]) / log_reg.coef_[0][1]

plt.clabel(contour, inline=1, fontsize=12)
plt.plot(left_right, boundary, "k--", linewidth=3)
plt.text(3.5, 1.5, "Not Iris-Virginica", fontsize=14, color="b", ha="center")
plt.text(6.5, 2.3, "Iris-Virginica", fontsize=14, color="g", ha="center")
plt.xlabel("Petal length", fontsize=14)
plt.ylabel("Petal width", fontsize=14)
plt.axis([2.9, 7, 0.8, 2.7])
save_fig("logistic_regression_contour_plot")
plt.show()
```

```
Saving figure logistic_regression_contour_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-83-c17bab459849> in <module>
     33 plt.ylabel("Petal width", fontsize=14)
     34 plt.axis([2.9, 7, 0.8, 2.7])
---> 35 save_fig("logistic_regression_contour_plot")
     36 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\logistic_regression_contour_plot.png'
```



In [84]:
```
X = iris["data"][:, (2, 3)]  # petal length, petal width
y = iris["target"]

softmax_reg = LogisticRegression(multi_class="multinomial",solver="lbfgs", C=10, random_state=42)
softmax_reg.fit(X, y)
```

Out[84]:
```
LogisticRegression(C=10, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='multinomial', n_jobs=None, penalty='l2',
                   random_state=42, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

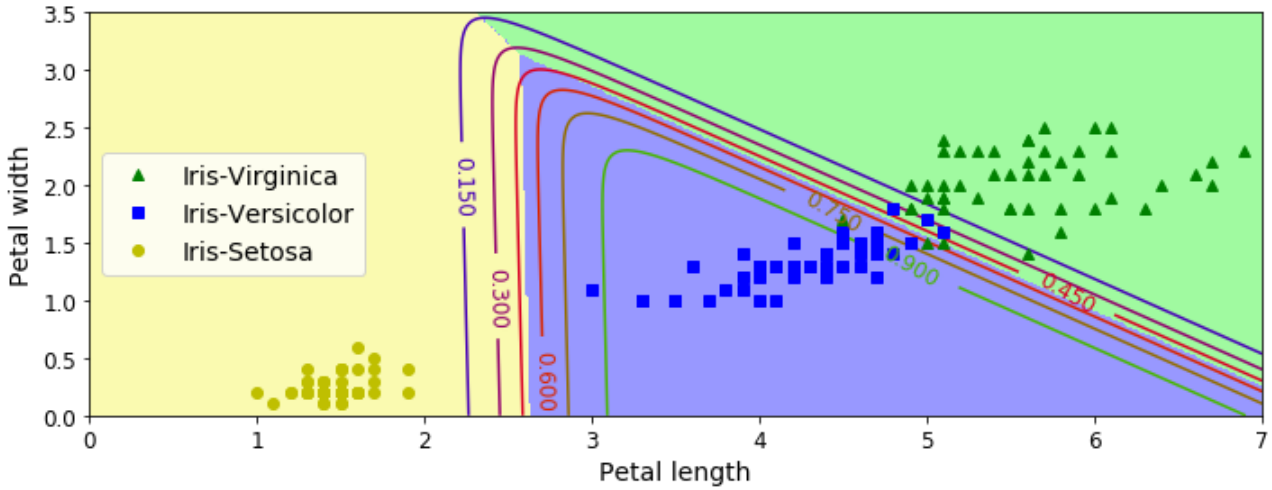
In [85]:

```
x0, x1 = np.meshgrid(
        np.linspace(0, 8, 500).reshape(-1, 1),
        np.linspace(0, 3.5, 200).reshape(-1, 1),
    )
X_new = np.c_[x0.ravel(), x1.ravel()]


y_proba = softmax_reg.predict_proba(X_new)
y_predict = softmax_reg.predict(X_new)

zz1 = y_proba[:, 1].reshape(x0.shape)
zz = y_predict.reshape(x0.shape)

plt.figure(figsize=(10, 4))
plt.plot(X[y==2, 0], X[y==2, 1], "g^", label="Iris-Virginica")
plt.plot(X[y==1, 0], X[y==1, 1], "bs", label="Iris-Versicolor")
plt.plot(X[y==0, 0], X[y==0, 1], "yo", label="Iris-Setosa")

from matplotlib.colors import ListedColormap
custom_cmap = ListedColormap(['#fafab0','#9898ff','#a0faa0'])

plt.contourf(x0, x1, zz, cmap=custom_cmap)
contour = plt.contour(x0, x1, zz1, cmap=plt.cm.brg)
plt.clabel(contour, inline=1, fontsize=12)
plt.xlabel("Petal length", fontsize=14)
plt.ylabel("Petal width", fontsize=14)
plt.legend(loc="center left", fontsize=14)
plt.axis([0, 7, 0, 3.5])
save_fig("softmax_regression_contour_plot")
plt.show()
```

```
Saving figure softmax_regression_contour_plot
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-85-b0dd46a31083> in <module>
     27 plt.legend(loc="center left", fontsize=14)
     28 plt.axis([0, 7, 0, 3.5])
---> 29 save_fig("softmax_regression_contour_plot")
     30 plt.show()

<ipython-input-19-a2c3eef30271> in save_fig(fig_id, tight_layout)
      8     if tight_layout:
      9         plt.tight_layout()
---> 10     plt.savefig(path, format='png', dpi=300)

~\Anaconda3\lib\site-packages\matplotlib\pyplot.py in savefig(*args, **kwargs)
    714 def savefig(*args, **kwargs):
    715     fig = gcf()
--> 716     res = fig.savefig(*args, **kwargs)
    717     fig.canvas.draw_idle()   # need this if 'transparent=True' to reset colors
    718     return res

~\Anaconda3\lib\site-packages\matplotlib\figure.py in savefig(self, fname, transparent, **kwargs)
   2178             self.patch.set_visible(frameon)
   2179
-> 2180         self.canvas.print_figure(fname, **kwargs)
   2181
   2182         if frameon:

~\Anaconda3\lib\site-packages\matplotlib\backend_bases.py in print_figure(self, filename, dpi, facecolor, edgecolor, orientatio
n, format, bbox_inches, **kwargs)
   2080                     orientation=orientation,
   2081                     bbox_inches_restore=_bbox_inches_restore,
-> 2082                     **kwargs)
   2083             finally:
   2084                 if bbox_inches and restore_bbox:

~\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py in print_png(self, filename_or_obj, metadata, pil_kwargs, *arg
s, **kwargs)
    528             renderer = self.get_renderer()
    529             with cbook._setattr_cm(renderer, dpi=self.figure.dpi), \
--> 530                     cbook.open_file_cm(filename_or_obj, "wb") as fh:
    531                 _png.write_png(renderer._renderer, fh,
    532                                self.figure.dpi, metadata=metadata)

~\Anaconda3\lib\contextlib.py in __enter__(self)
    110         del self.args, self.kwds, self.func
    111         try:
--> 112             return next(self.gen)
    113         except StopIteration:
    114             raise RuntimeError("generator didn't yield") from None

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in open_file_cm(path_or_file, mode, encoding)
    445 def open_file_cm(path_or_file, mode="r", encoding=None):
```

```
    446     r"""Pass through file objects and context-manage `.PathLike`\s."""
--> 447     fh, opened = to_filehandle(path_or_file, mode, True, encoding)
    448     if opened:
    449         with fh:

~\Anaconda3\lib\site-packages\matplotlib\cbook\__init__.py in to_filehandle(fname, flag, return_opened, encoding)
    430             fh = bz2.BZ2File(fname, flag)
    431         else:
--> 432             fh = open(fname, flag, encoding=encoding)
    433         opened = True
    434     elif hasattr(fname, 'seek'):

FileNotFoundError: [Errno 2] No such file or directory: 'D:\\My ML Simulations\\Linear Regression\\images\\training_linear_models\\softmax_regression_contour_plot.png'
```



In [87]:
```
softmax_reg.predict([[5, 2]])
```

Out[87]:
```
array([2])
```

In [88]:
```
softmax_reg.predict_proba([[5, 2]])
```

Out[88]:
```
array([[6.38014896e-07, 5.74929995e-02, 9.42506362e-01]])
```