|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | In [1]:  **import** **numpy** **as** **np**  **from** **matplotlib** **import** pyplot **as** plt  **from** **numpy.random** **import** normal  **from** **scipy.stats** **import** norm  **from** **math** **import** sqrt  np.random.seed(40)  In [2]:  x = np.linspace(-1, 1, 10)  line = []  **for** i **in** x:  i = -0.3 + 0.5 \* i + normal(0, 0.2)  line.append(i)  plt.scatter(x, line)  plt.plot(x, -0.3 + 0.5 \* x, c="red")  Out[2]:  [<matplotlib.lines.Line2D at 0x7f07f83df588>]  In [3]:  **def** linear(x, w0, w1):  **return** w0 + w1\*x    **def** gaussian(y, x, w0, w1):  sd = 0.2  cons= sqrt(np.pi \* 2) \* sd  **return** cons \* np.exp(-0.5 \* ((y - linear(x, w0, w1))/sd) \*\* 2)  In [4]:  w0, w1 = np.meshgrid(np.linspace(-1, 1, 100), np.linspace(-1, 1, 100))  grid\_0 = norm.pdf(w0, 0, sqrt(0.5))  grid\_1 = norm.pdf(w1, 0, sqrt(0.5))  prior = grid\_0 \* grid\_1  plt.contourf(w0, w1, prior, cmap="jet")  Out[4]:  <matplotlib.contour.QuadContourSet at 0x7f07f7e25550>  In [5]:  fig, axs = plt.subplots(10, 2, figsize=(3, 10))  count = 0  **for** i, j **in** zip(x, line):  mle = gaussian(j, i, w0, w1)  posterior = prior \* mle  prior = posterior  axs[count, 0].contourf(w0, w1, mle, cmap="jet")  axs[count, 1].contourf(w0, w1, posterior, cmap="jet")  count = count+ 1  In [7]:  fig, axes = plt.subplots(1, 2, figsize=(8, 4))  axes[0].contourf(w0, w1, mle, cmap="jet")  axes[0].scatter(-0.3, 0.5, c="blue", marker="+")  axes[1].contourf(w0, w1, posterior, cmap="jet")  https://lh3.googleusercontent.com/a-/AOh14Gj20VmwY5zZeyEhuWPZqkjoKiWIP_oHhE3PC-gR=s40axes[1].scatter(-0.3, 0.5, c="blue", marker="+")  Out[7]:  <matplotlib.collections.PathCollection at 0x7f07f741dc50>   |  |  | | --- | --- | |  |  | |  |