Generalization and stability

Moritz Hardt

CS 189 Fall 2018

Berkeley
UNIVERSITY OF CALIFORNIA

Announcements

Midterm exam during class 9:30–11a, Thursday 10/18

Review sessions Tuesday 10/16

Check Piazza for details!

Extra office hour after class in SDH 722

Midterm details

You will not be able to ask questions during the exam

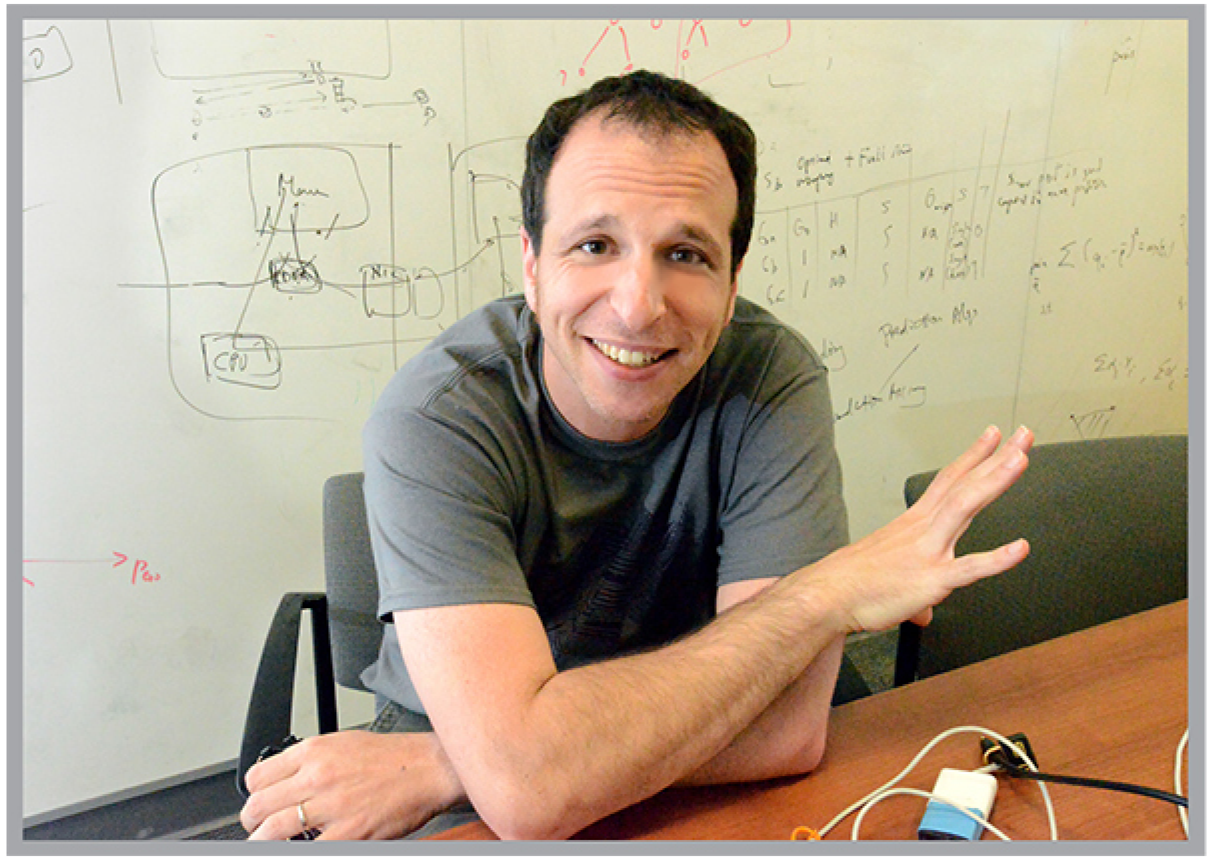
You are allowed 1 double-sided letter format, **handwritten** cheat sheet

Remember to bring your Student ID with you

Sections will be cancelled on Thu 10/18 and Fri 10/19

Additional midterm resources have been posted on Piazza

The whole class so far



"SGD is all you need, man."

"Pretty much any loss function works."

What's up next

Today: **Generalization and stability**

After the midterm: **Non-convex optimization and deep learning**

November: **Machine learning as if people mattered**

- fairness, societal impact, understanding risks
- loosely based on fairmlbook.org

Recap: Slightly more formally…

You have labeled examples $z_1, \ldots, z_n$ where $z_i = (x_i, y_i)$

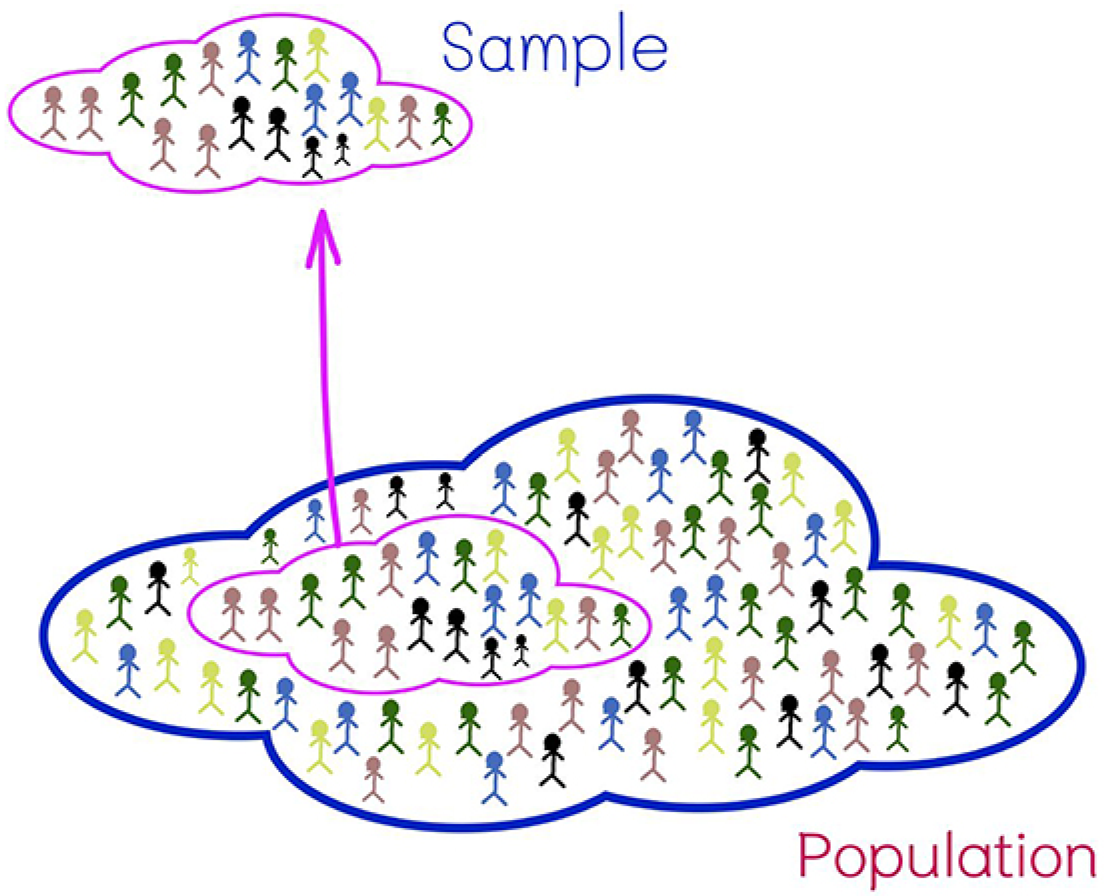
We can solve $\min_w \frac{1}{n} \sum_{i=1}^{n} \text{loss}(w^\top x_i, y_i)$

<u>Stochastic Gradient Descent (SGD)</u>

Start from initial parameters w_0. Repeat:

- Pick random example index i
- Update
  $w_{t+1} \leftarrow w_t - \alpha x_i \nabla_p \text{loss}(p, y)|_{p=w_t^\top x_i}$

Where does your data come from?



Sample represents a population

The goal of learning is to learn about the population, not the sample!

Common math assumption:
Population is represented by a *distribution*

What this means for classification

We assume examples $z_1, \ldots, z_n$ are drawn *independently and identically* from an unknown distribution

Our goal: $\min_w \mathbb{E}[\text{loss}(w^\top x, y)]$

We want to classify well on the population population

Solving $\min_w \sum_{i=1}^n \frac{1}{n} \text{loss}(w^\top x_i, y_i)$ only guarantees we're good on the sample

How can we connect the two?

Let's give these names

Risk: $R(w) = \mathbb{E}[\text{loss}(w^\top x, y)]$

Empirical risk: $R_S(w) = \frac{1}{n} \sum_{i=1}^n \text{loss}(w^\top x_i, y_i)$,
where $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ is the sample.

Risk minimization: $\min_w R(w)$

Empirical risk minimization: $\min_w R_S(w)$

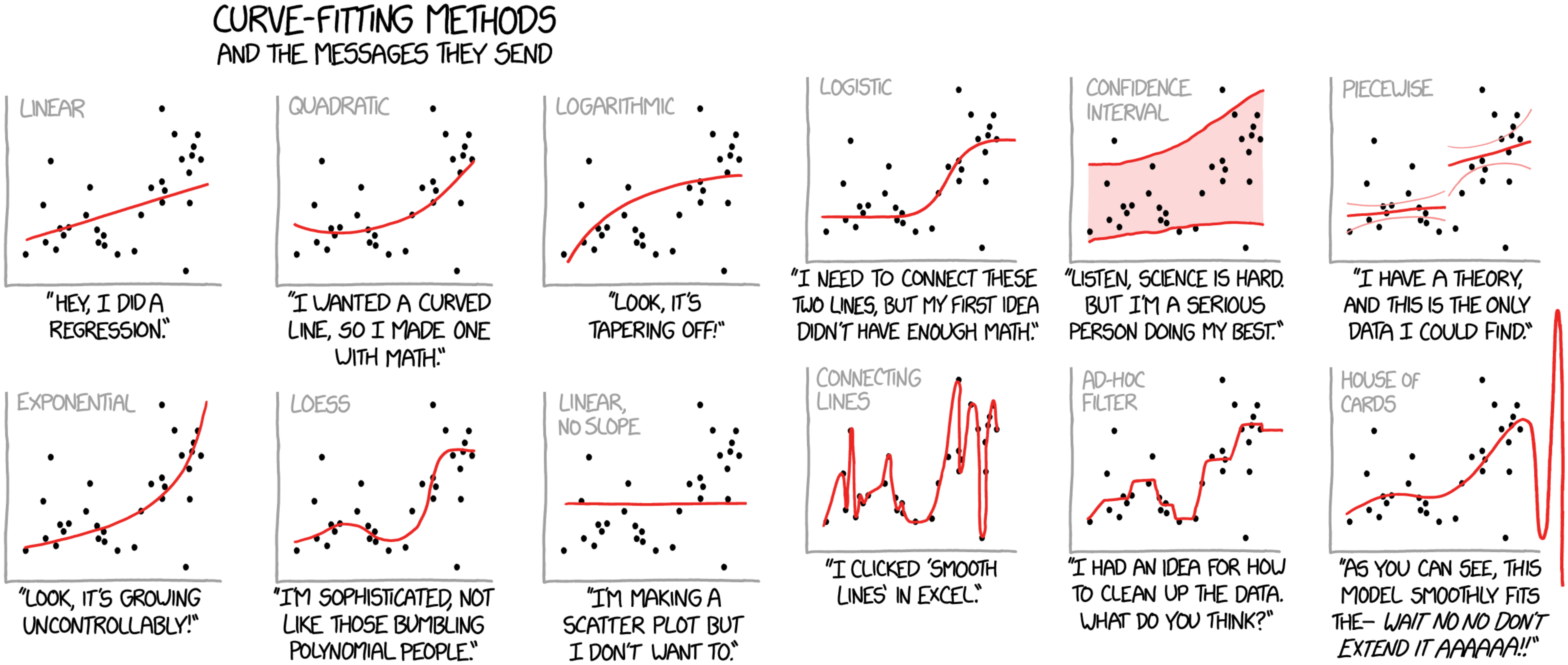
Risk	Empirical Risk
How well are you doing on an unknown examples	Empirical risk: How well are you doing on an known examples
Also called **test error**	Also called **training error**
Risk minimization is what we actually want!	Empirical risk minimization is what we can actually do via optimization.

The fundamental leap of faith

By minimizing empirical risk we hope that we also minimize risk!

This is called **generalization**

Failure to generalize sometimes called **overfitting**


CURVE-FITTING METHODS
AND THE MESSAGES THEY SEND
LINEAR
"HEY, I DID A REGRESSION."
QUADRATIC
"I WANTED A CURVED LINE, SO I MADE ONE WITH MATH."
LOGARITHMIC
"LOOK, IT'S TAPERING OFF!"
LOGISTIC
"I NEED TO CONNECT THESE TWO LINES, BUT MY FIRST IDEA DIDN'T HAVE ENOUGH MATH."
CONFIDENCE INTERVAL
"LISTEN, SCIENCE IS HARD. BUT I'M A SERIOUS PERSON DOING MY BEST."
PIECEWISE
"I HAVE A THEORY, AND THIS IS THE ONLY DATA I COULD FIND."
EXPONENTIAL
"LOOK, IT'S GROWING UNCONTROLLABLY!"
LOESS
"I'M SOPHISTICATED, NOT LIKE THOSE BUMBLING POLYNOMIAL PEOPLE."
LINEAR, NO SLOPE
"I'M MAKING A SCATTER PLOT BUT I DON'T WANT TO."
CONNECTING LINES
"I CLICKED 'SMOOTH LINES' IN EXCEL."
AD-HOC FILTER
"I HAD AN IDEA FOR HOW TO CLEAN UP THE DATA. WHAT DO YOU THINK?"
HOUSE OF CARDS
"AS YOU CAN SEE, THIS MODEL SMOOTHLY FITS THE– WAIT NO NO DON'T EXTEND IT AAAAAA!!"

Generalization gap:

$$\epsilon_{\text{gen}}(w) := R(w) - R_S(w)$$

Fundamental theorem of Machine Learning

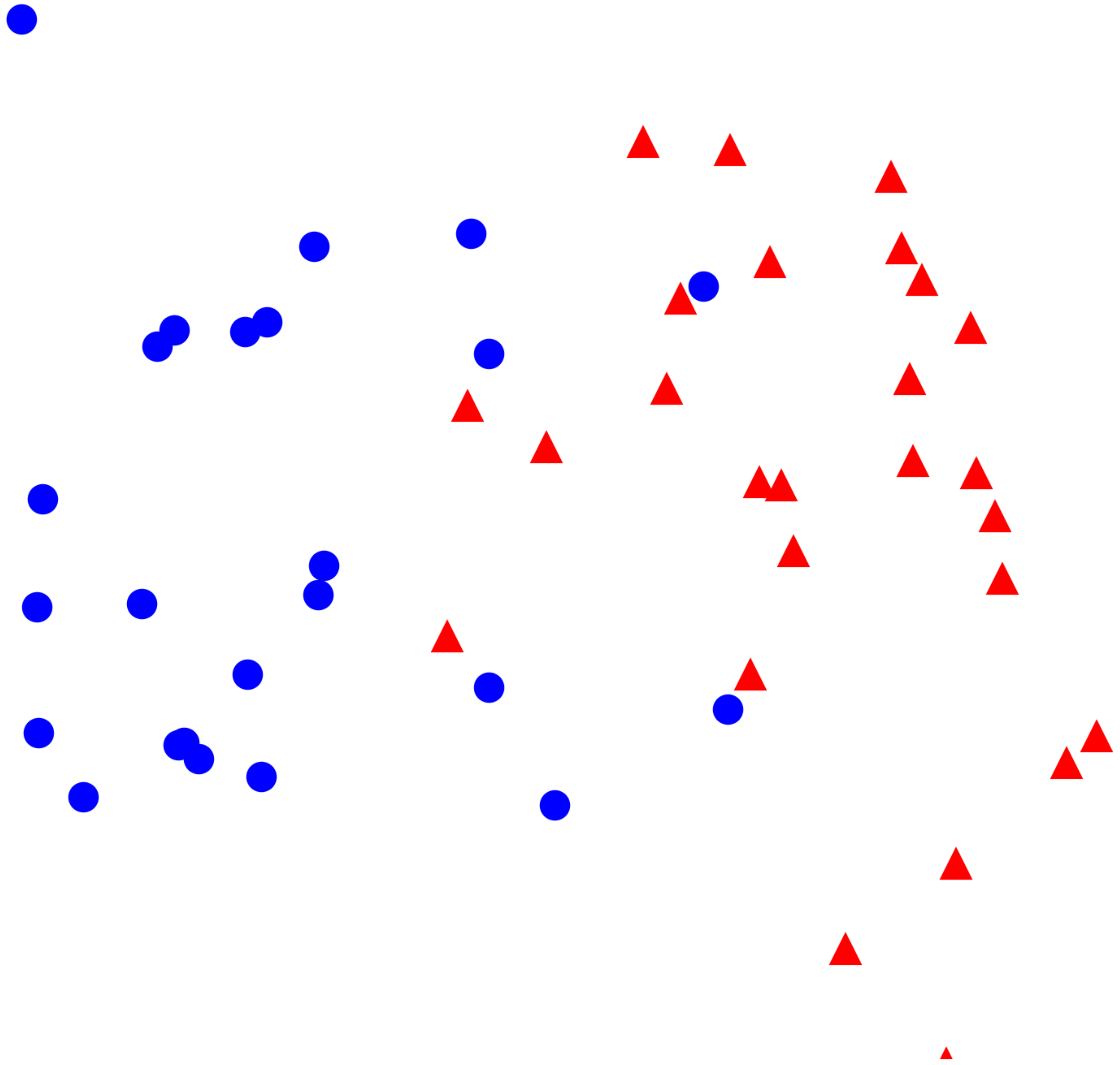
$$R(w) = R_S(w) + \epsilon_{\text{gen}}(w)$$

Proof might be a midterm problem!

How can we make sure $R(w) - R_S(w)$ is small?

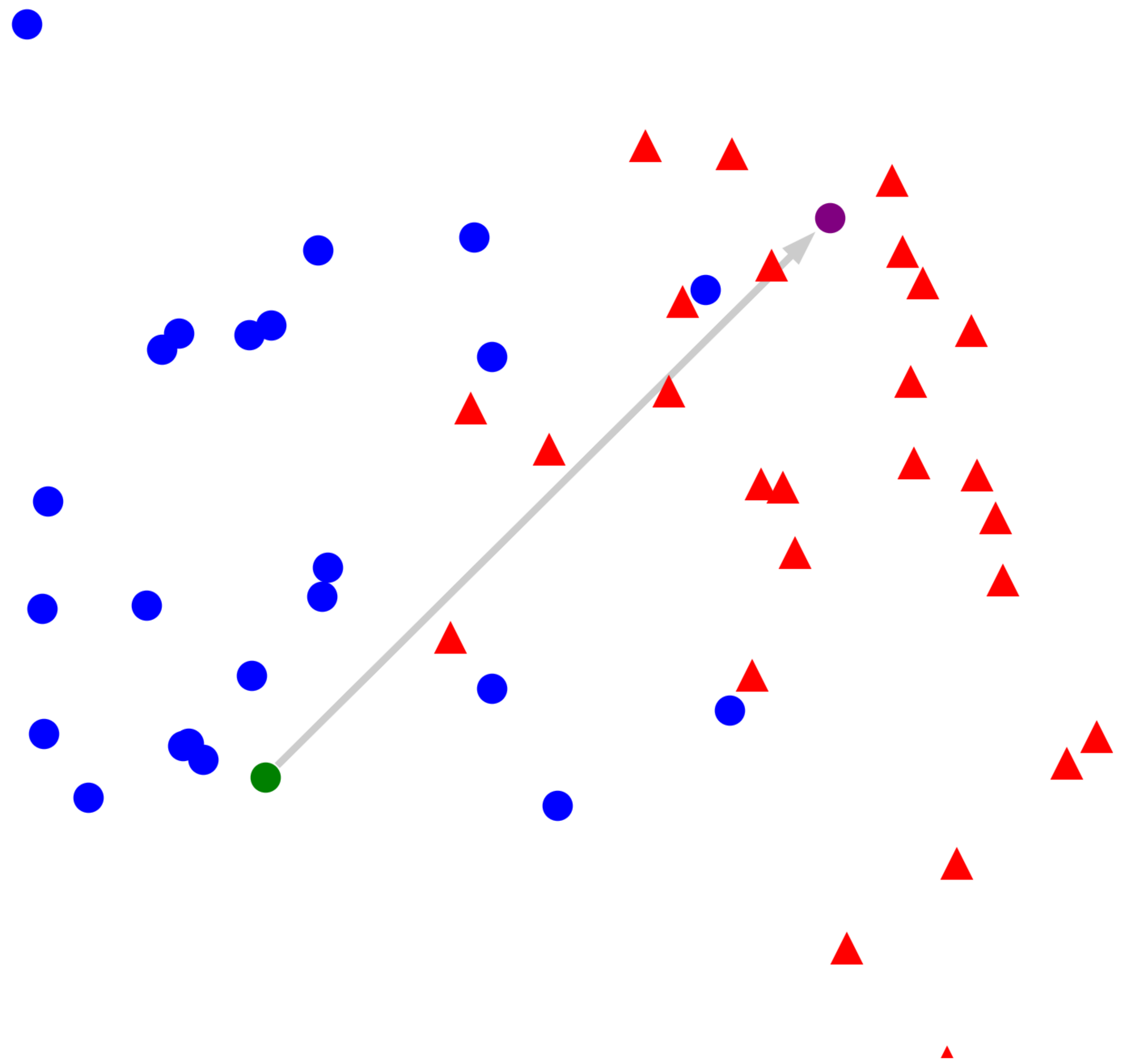
This lecture: **Robustness of the learning algorithm**

The idea behind robustness



Suppose we want to classify triangles from dots

The idea behind robustness



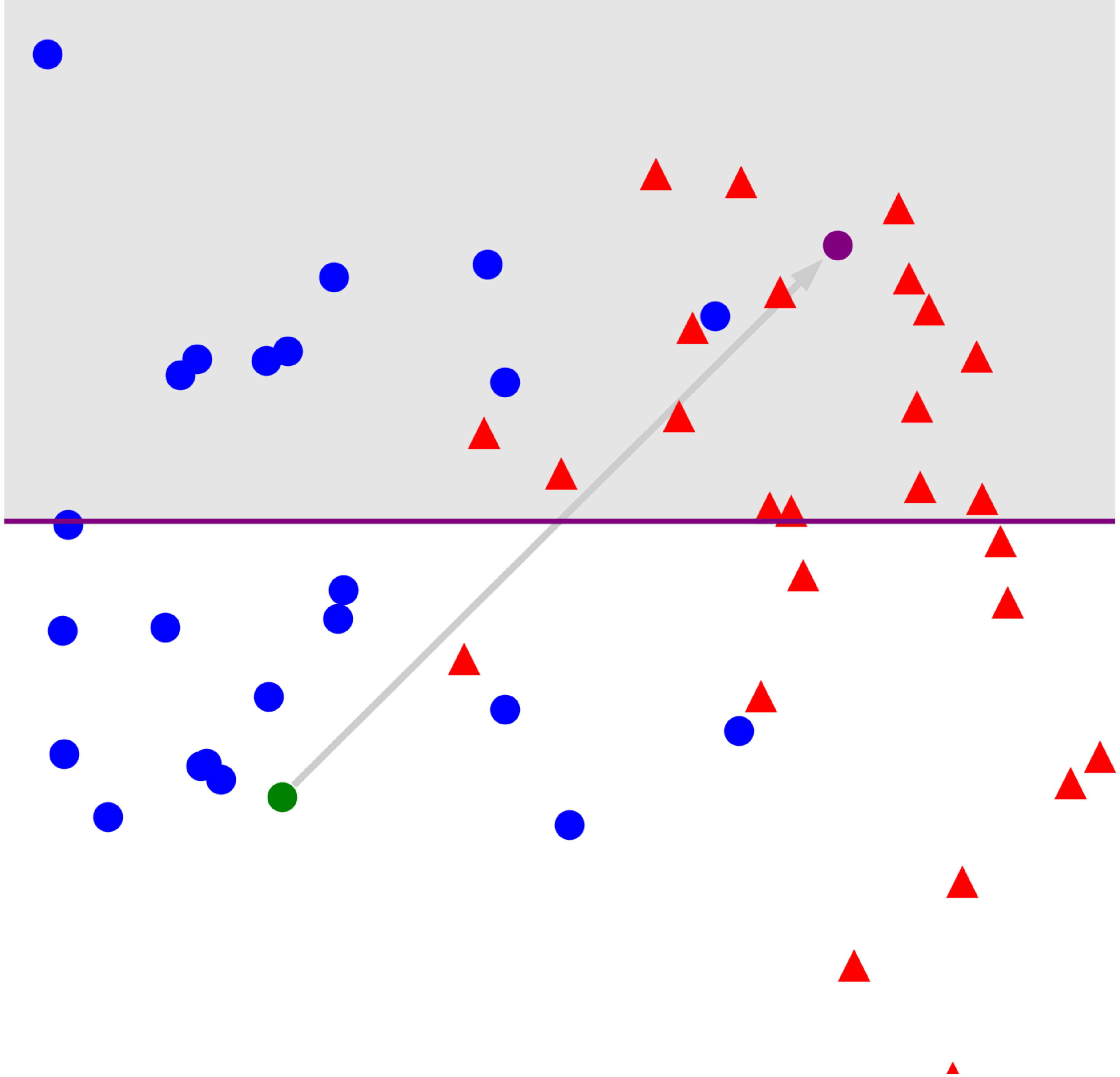
Intuitively, a learning algorithm shouldn't care much if you change one of the training examples.

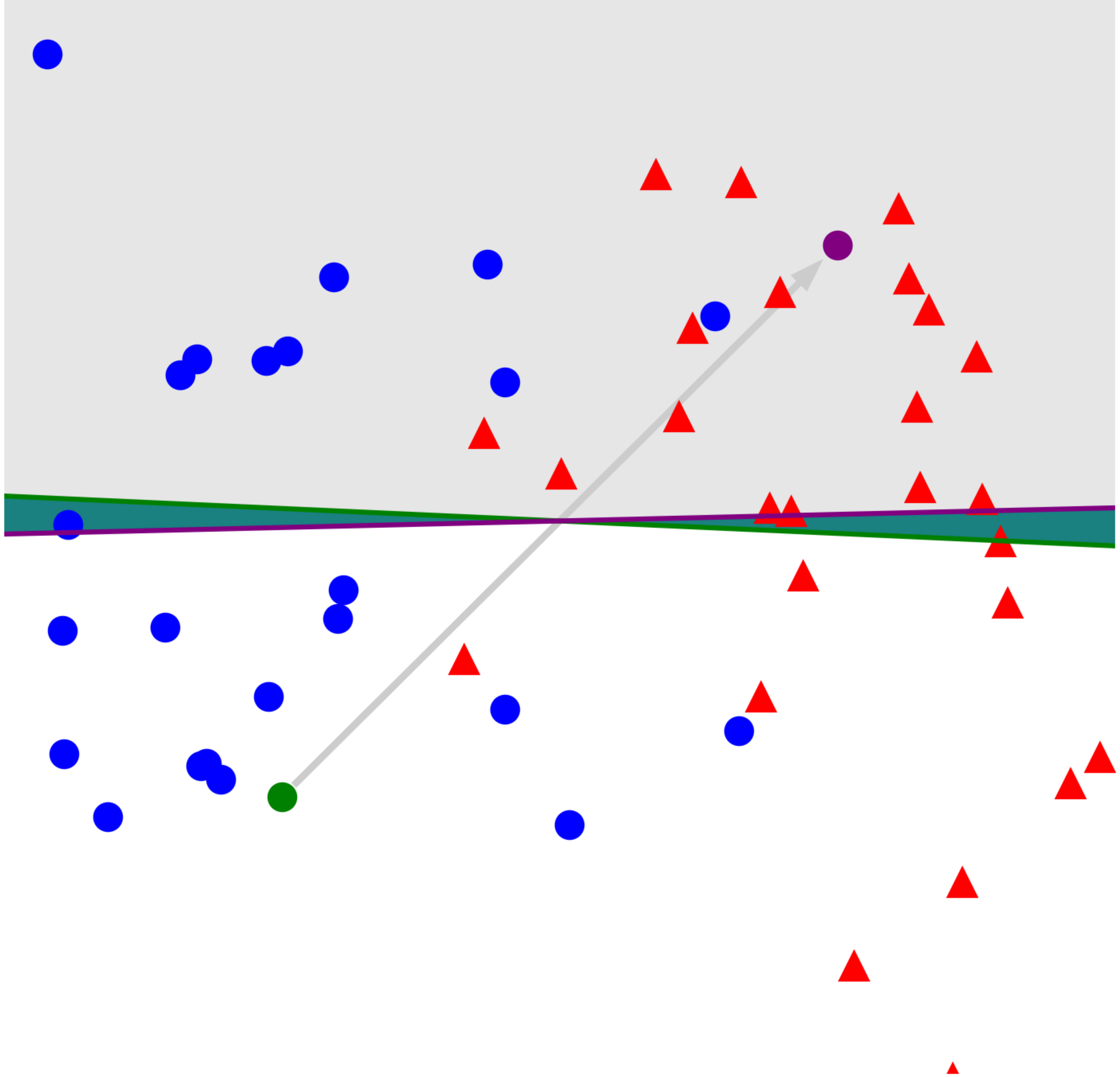
Turns out, if this is true, then the algorithm also generalizes.

Example: SGD for linear classification

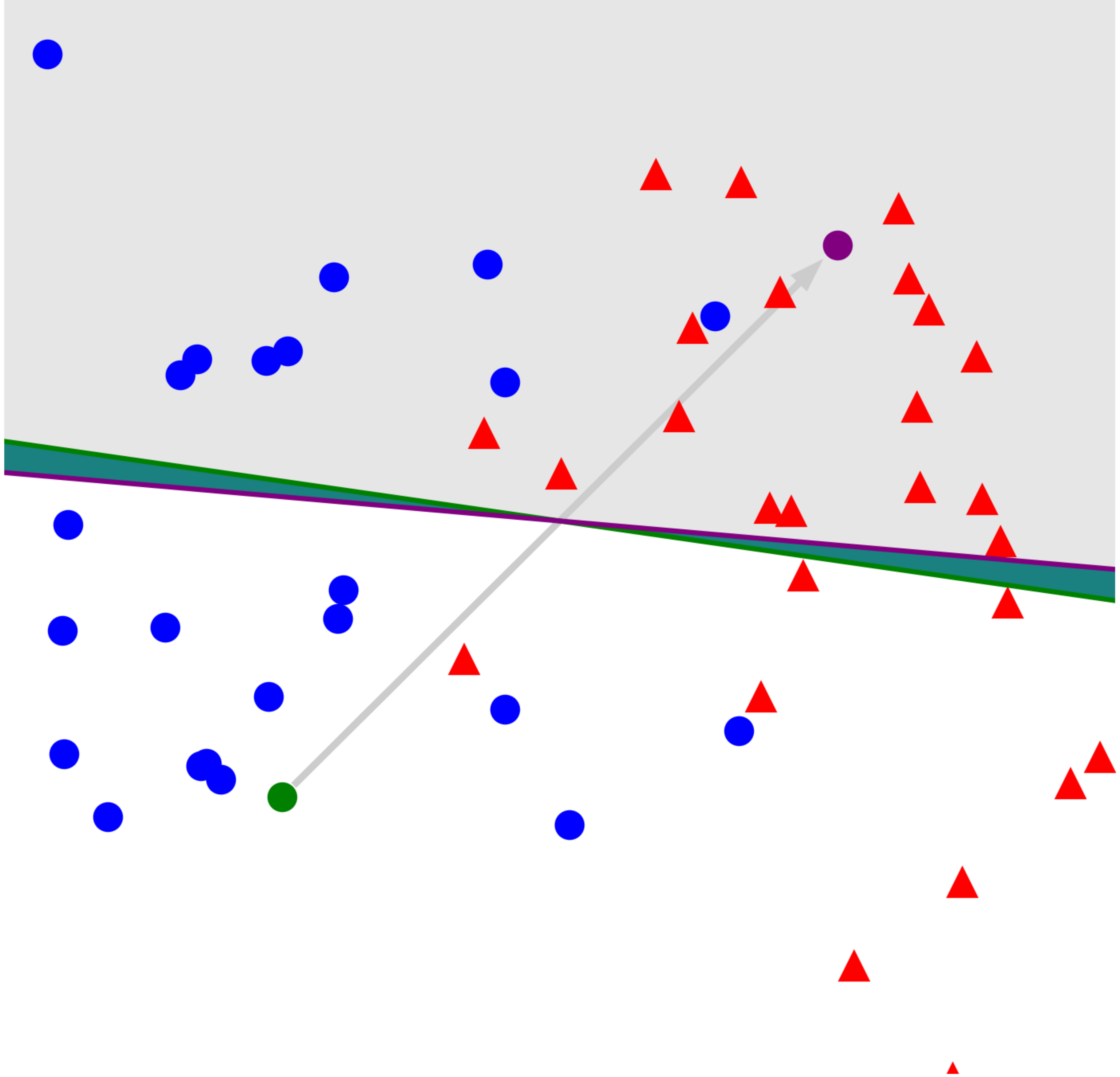
Let's see how stable SGD is in a simulation

Linear classification, squared loss (anything would work really)

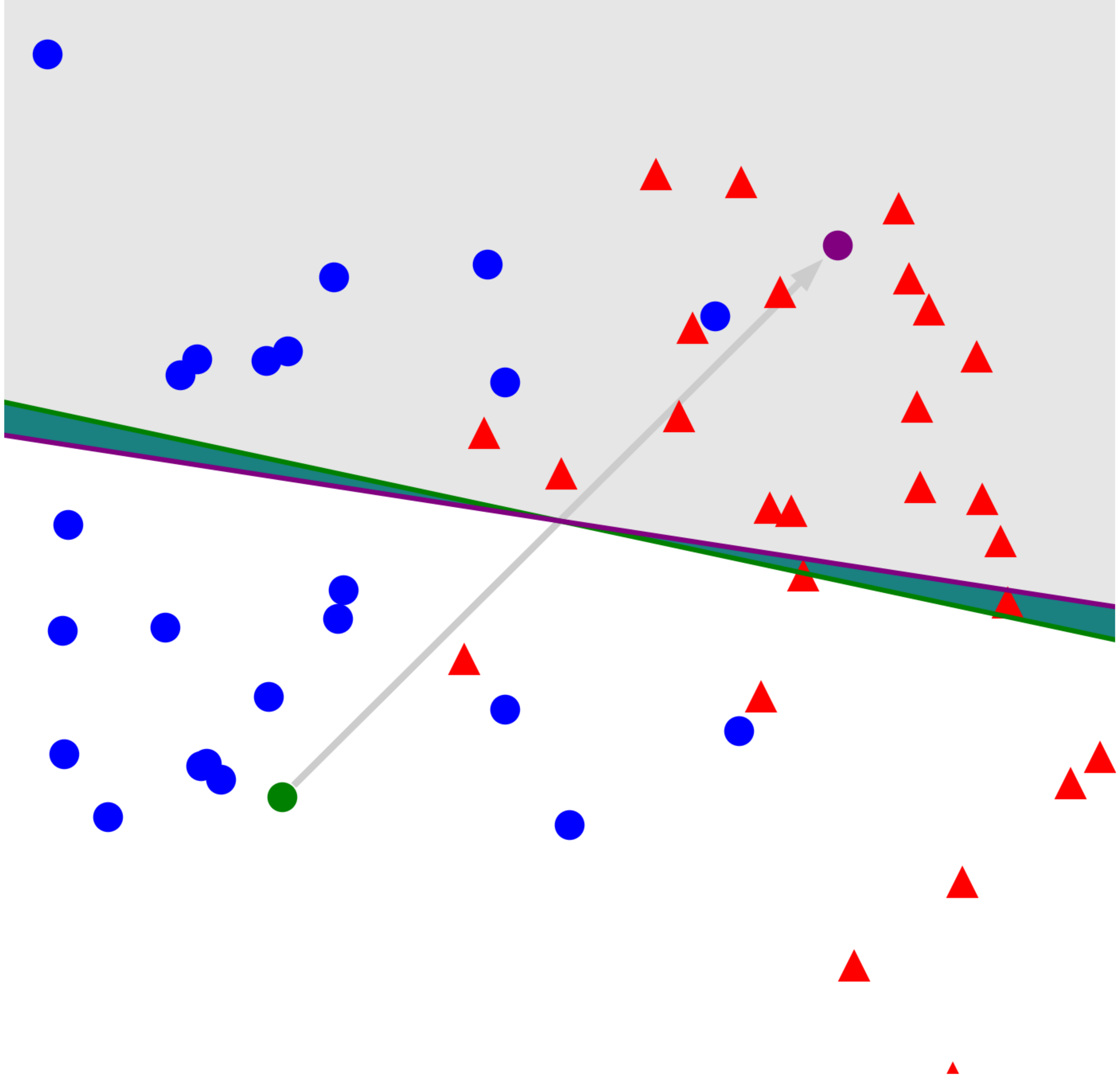


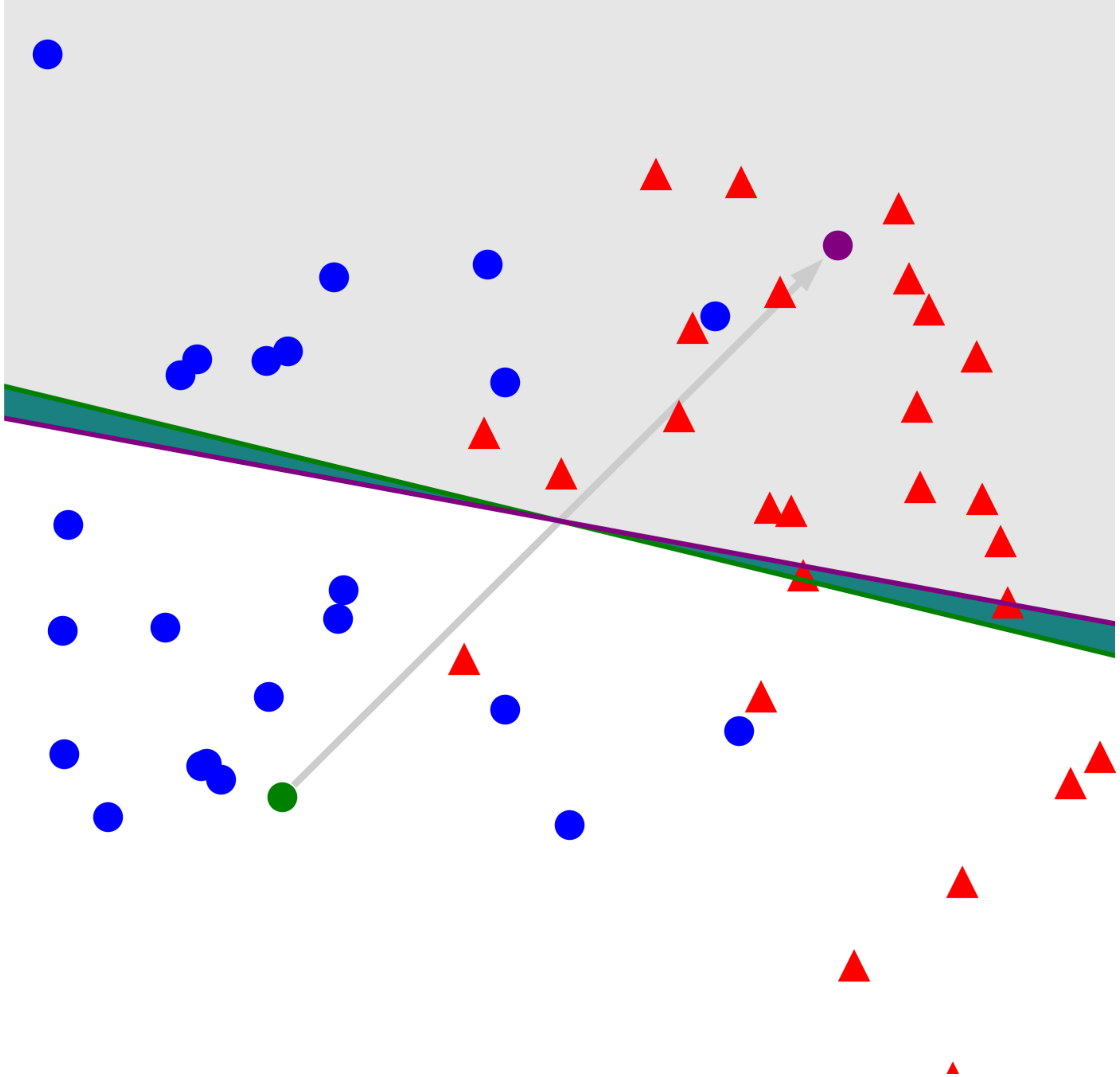


Step 2/30

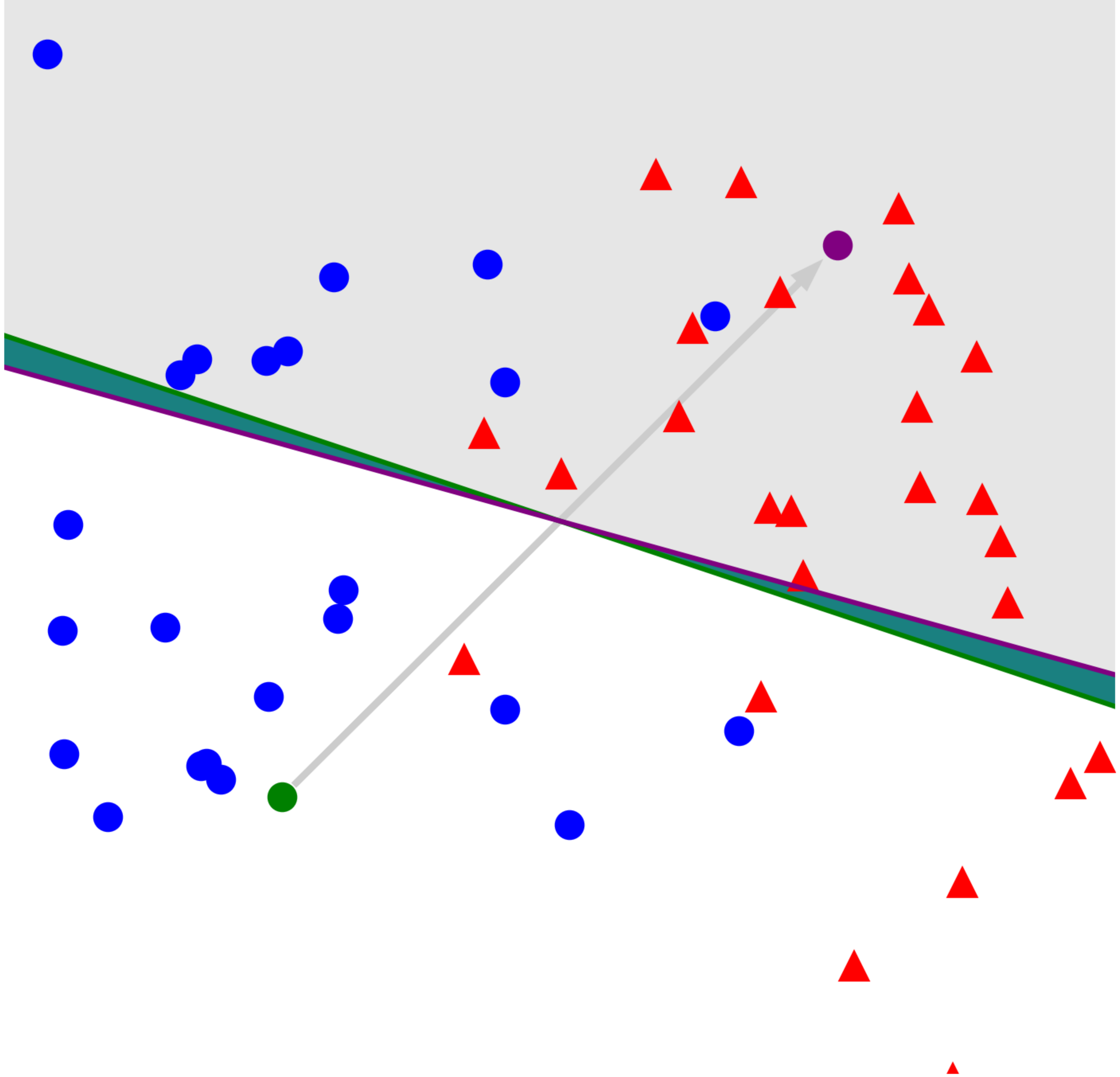


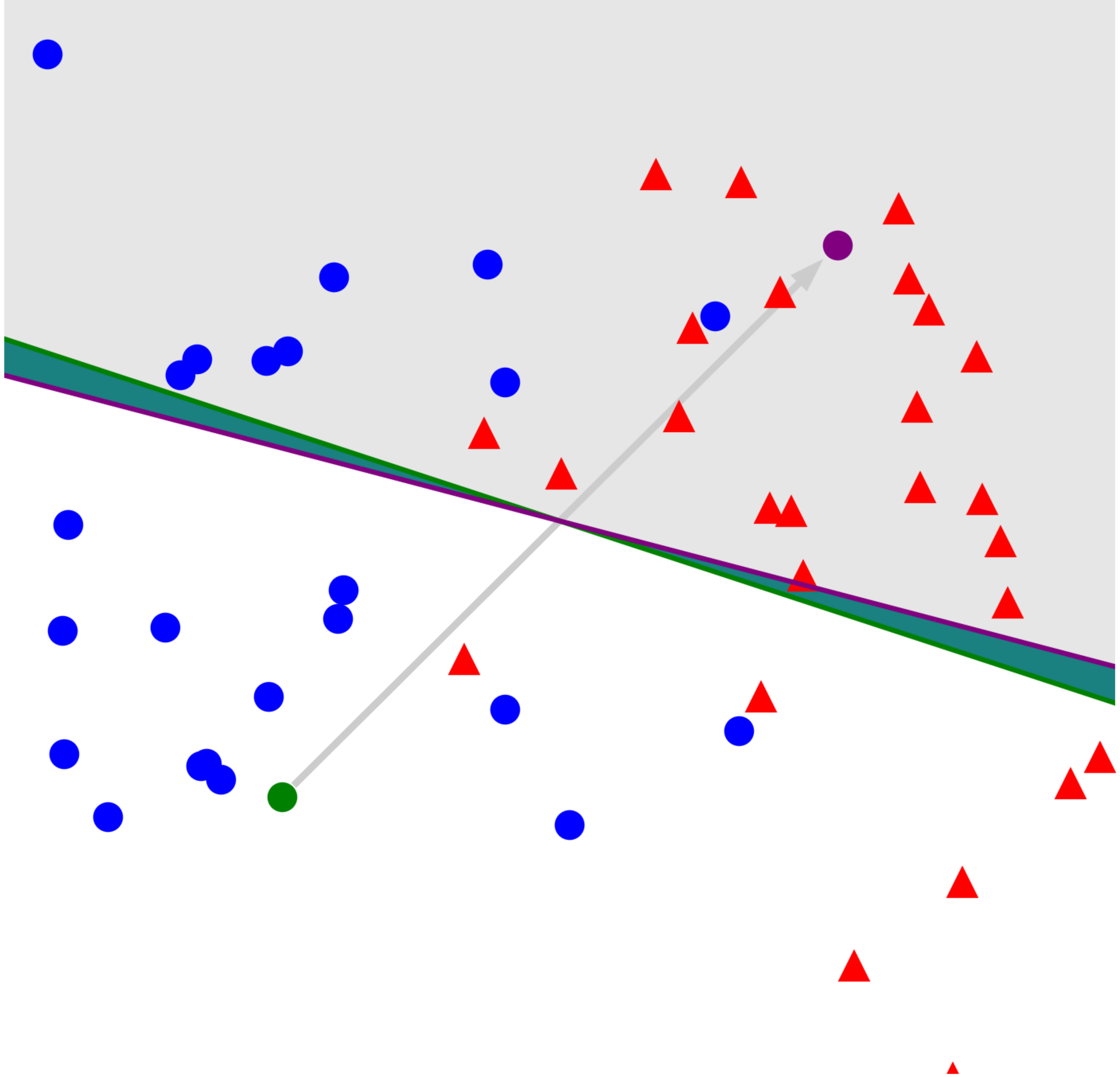
Step 3/30



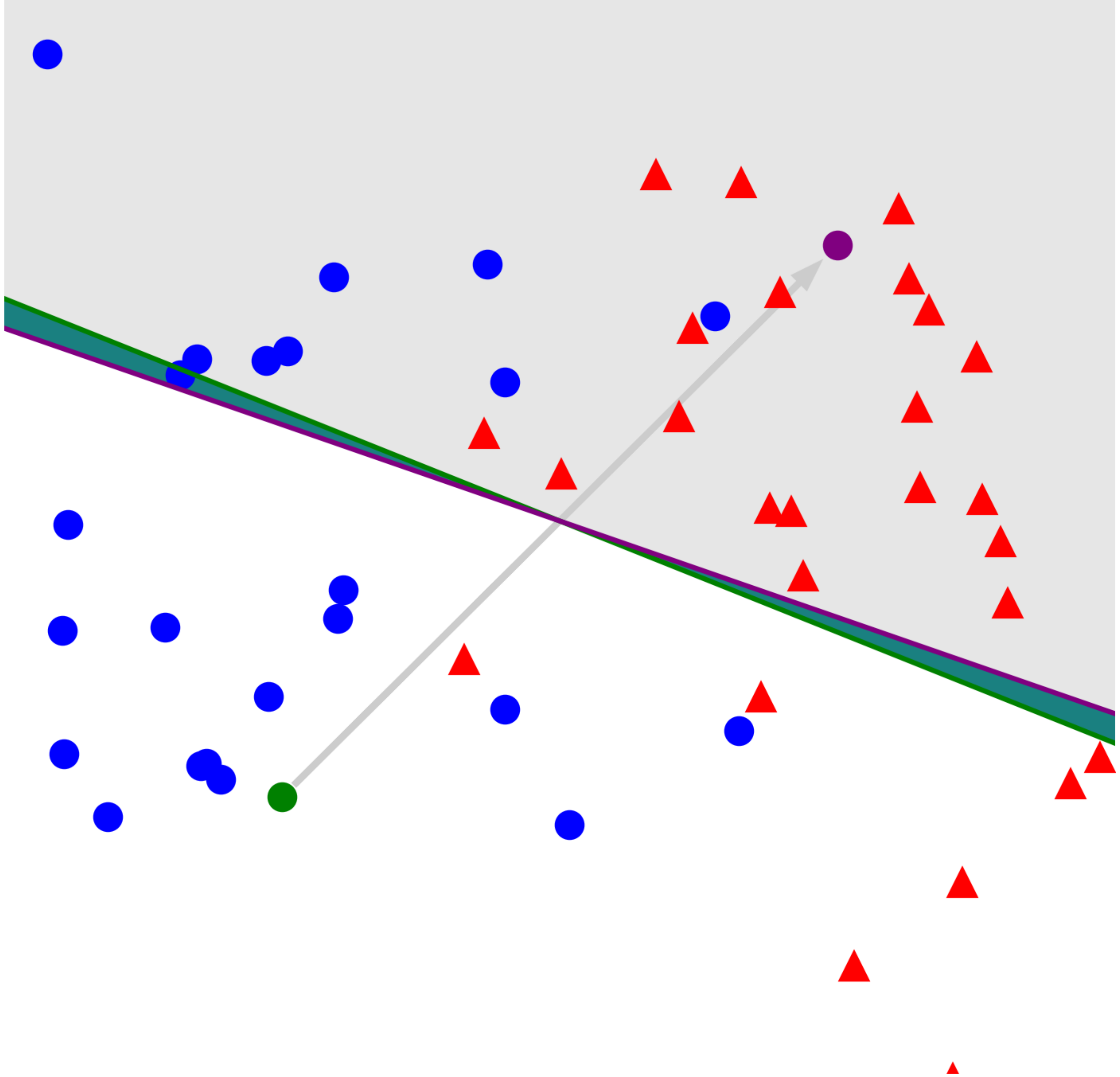


Step 5/30

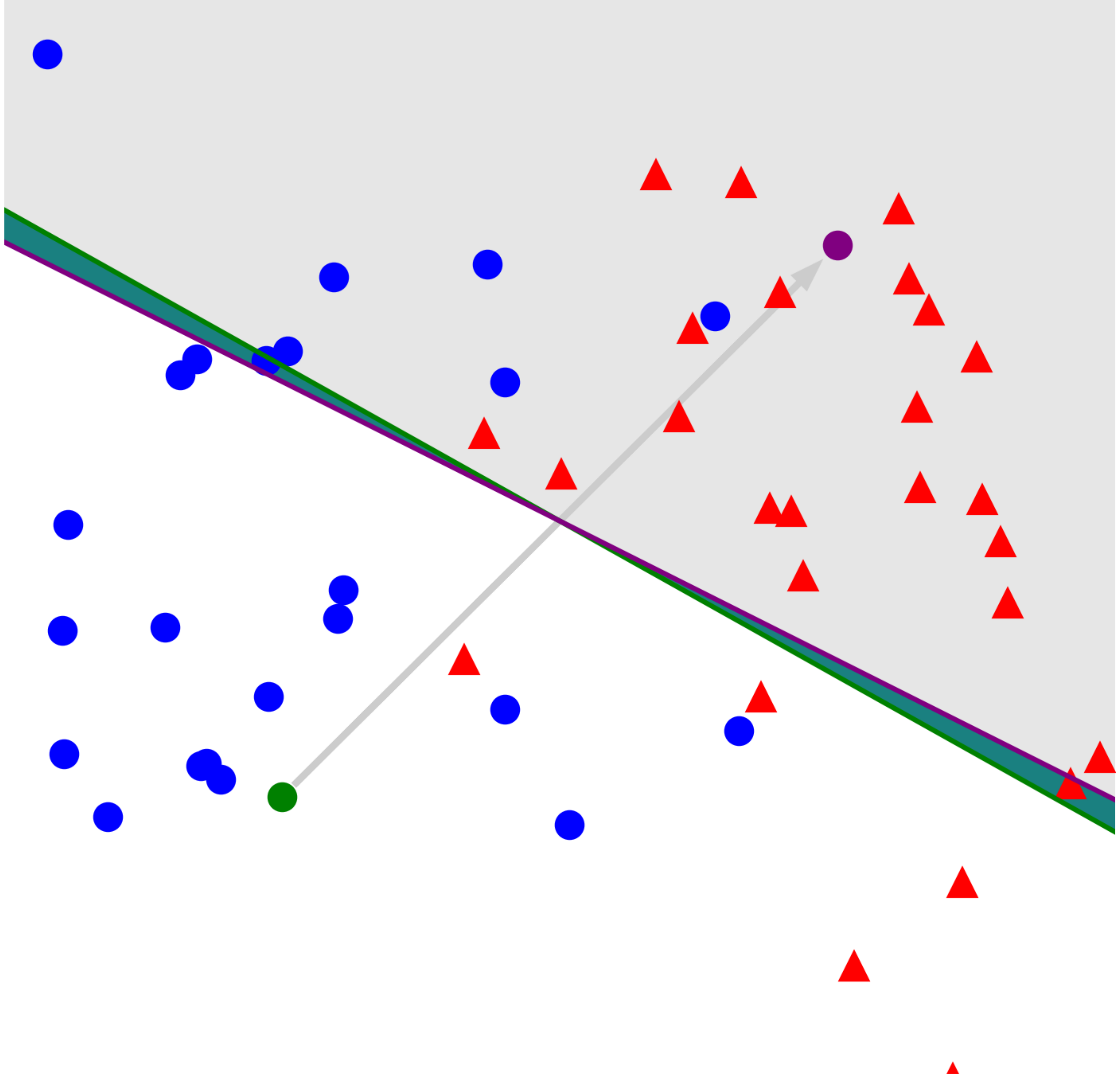




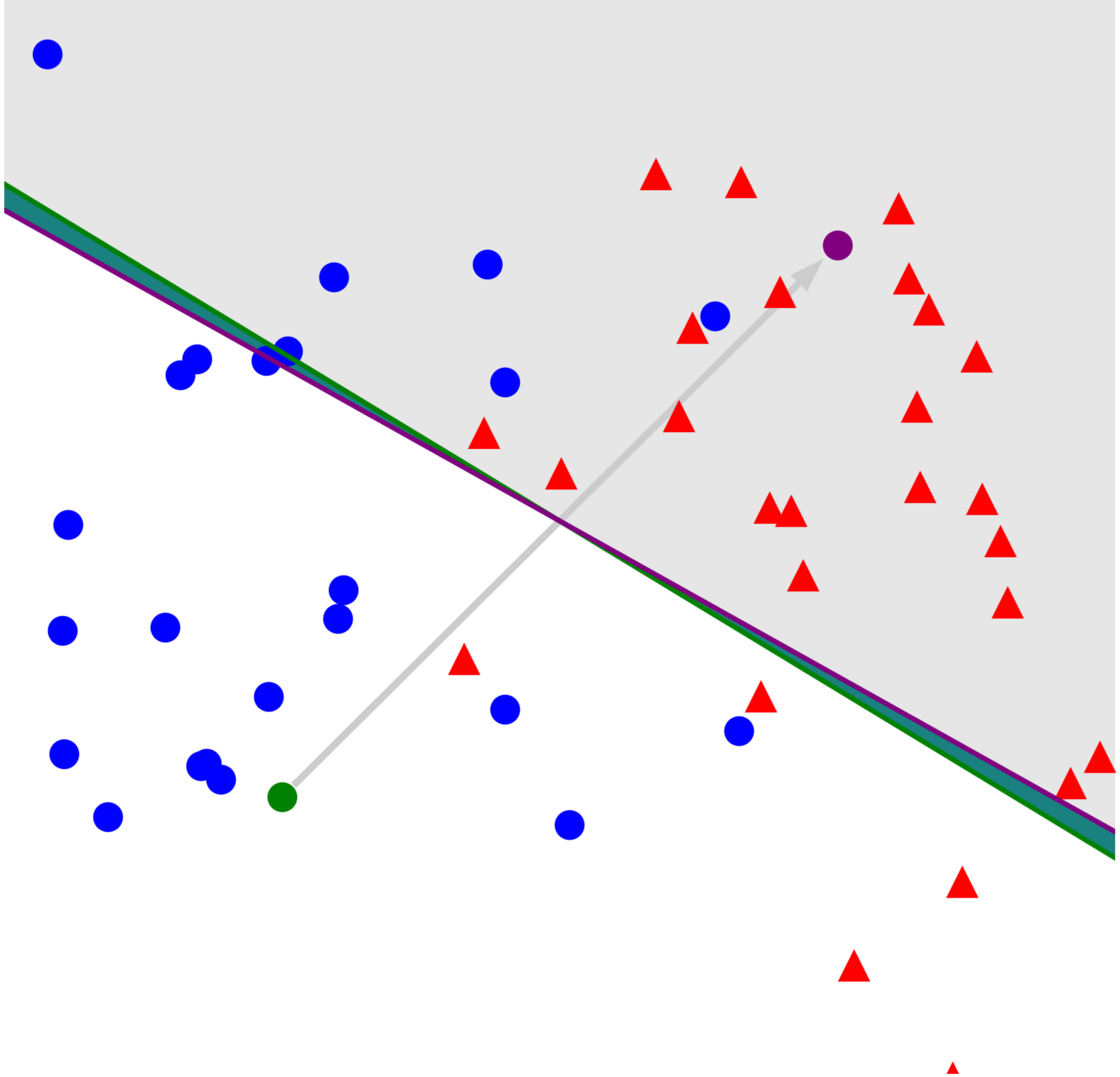
Step 7/30



Step 8/30

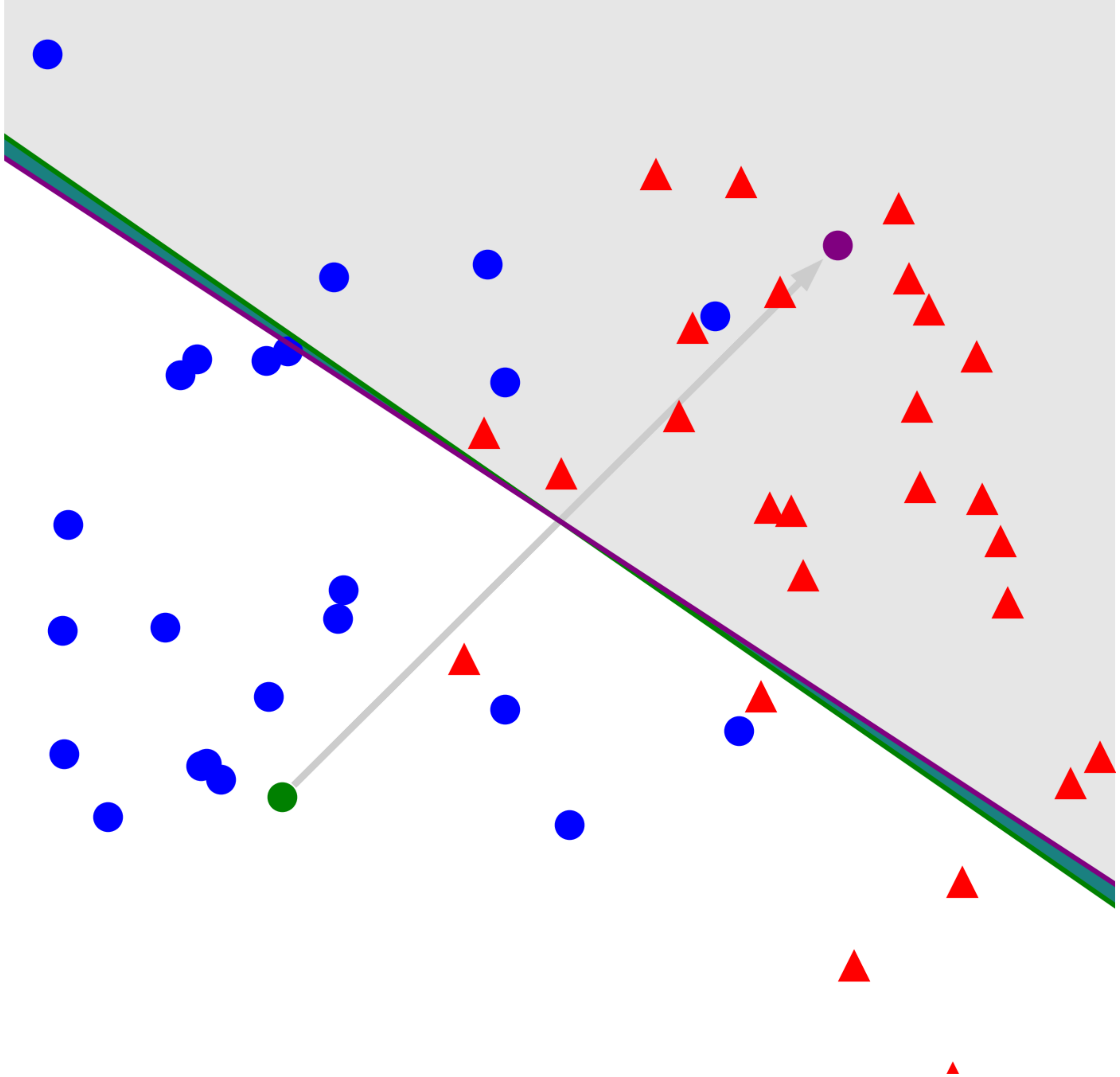


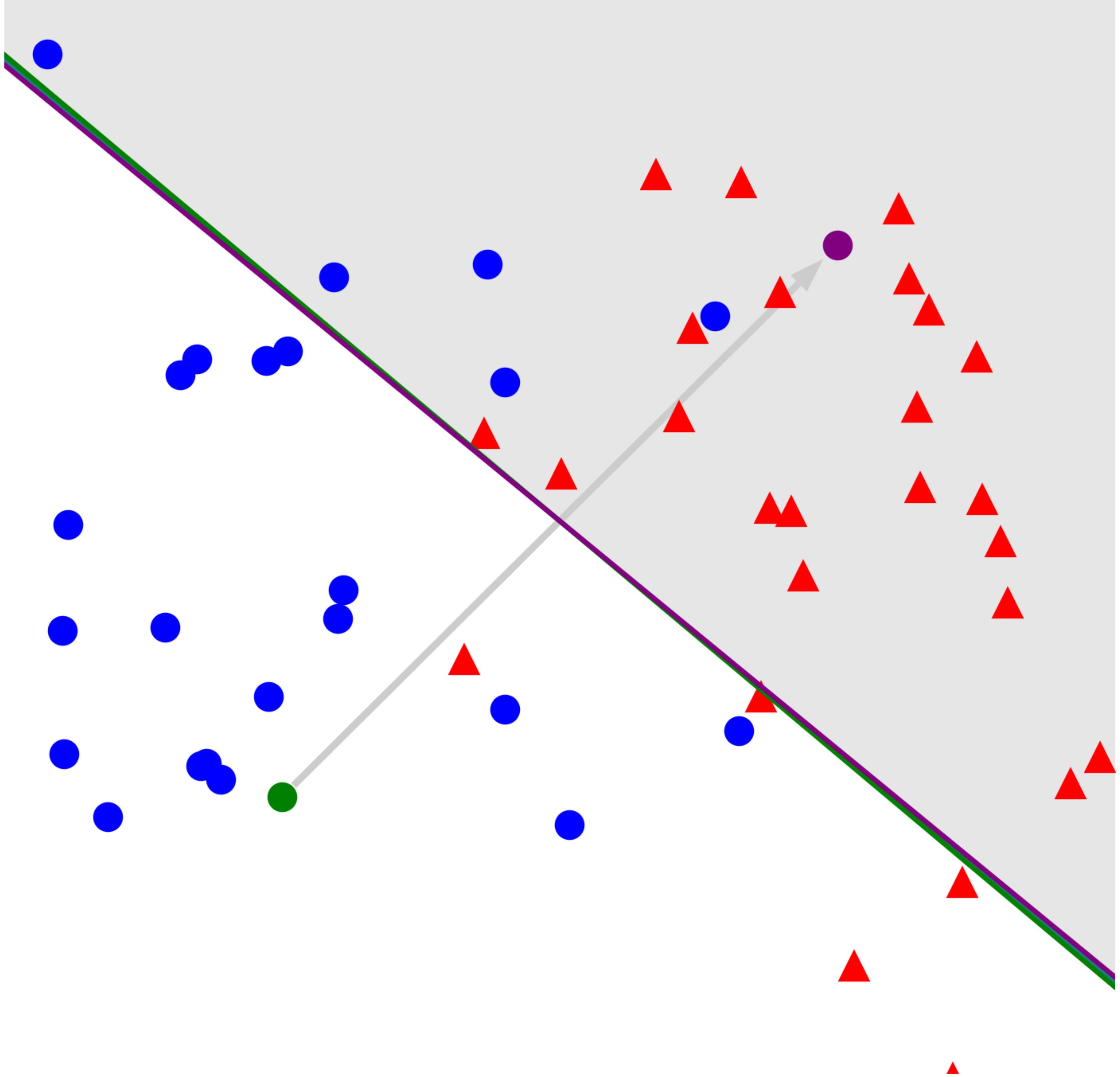
Step 9/30



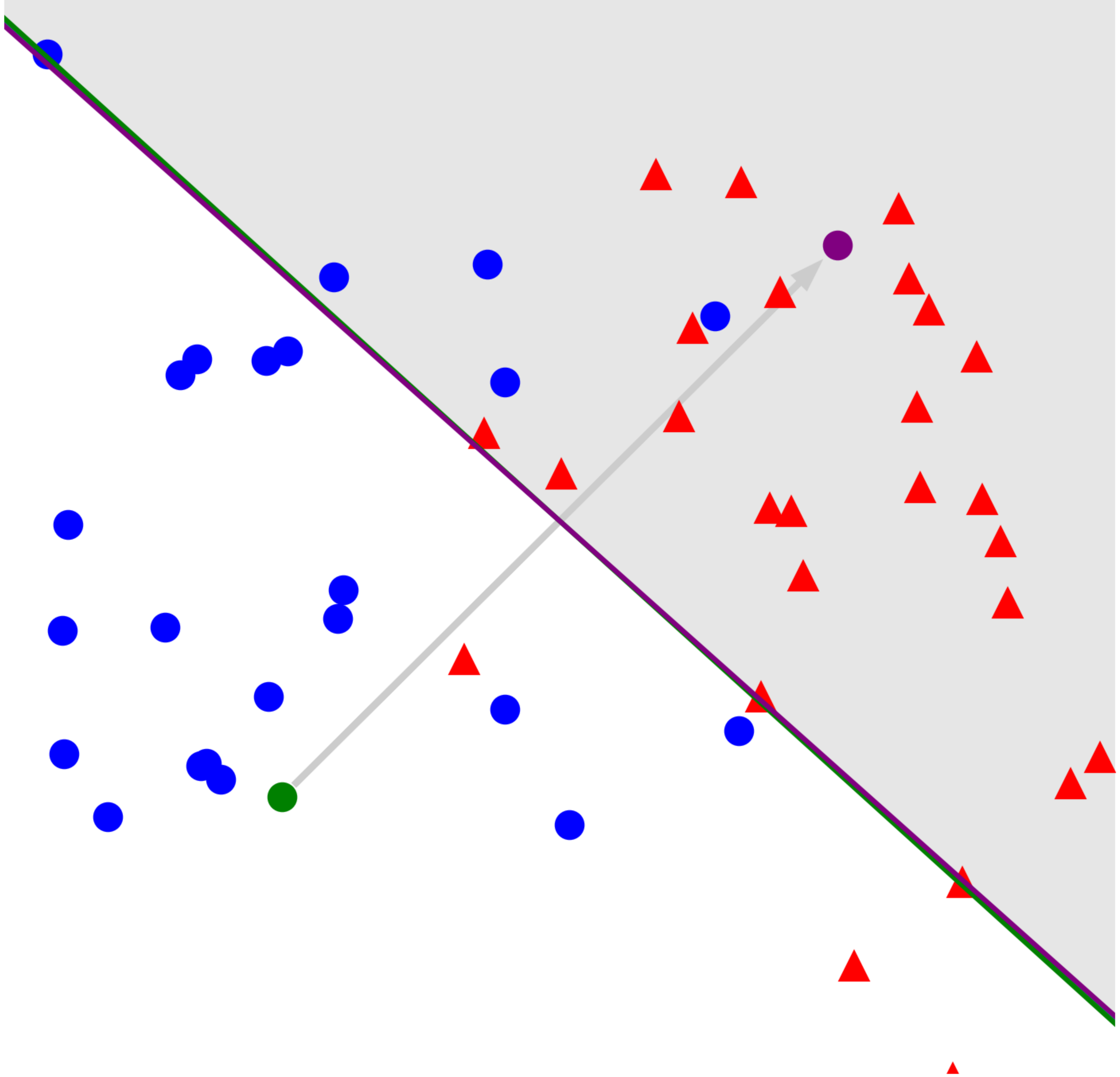
Step 10/30

Step 11/30

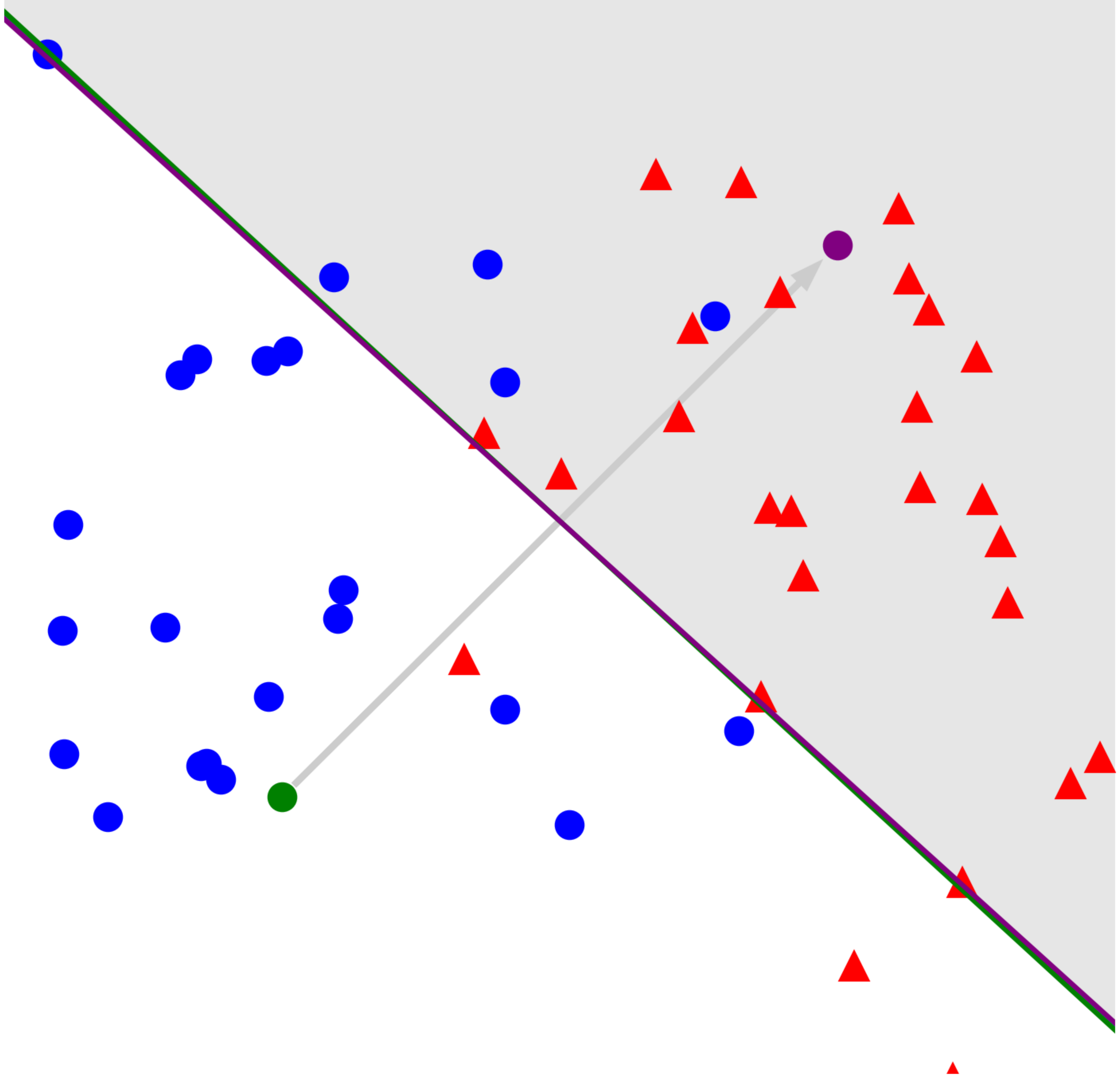




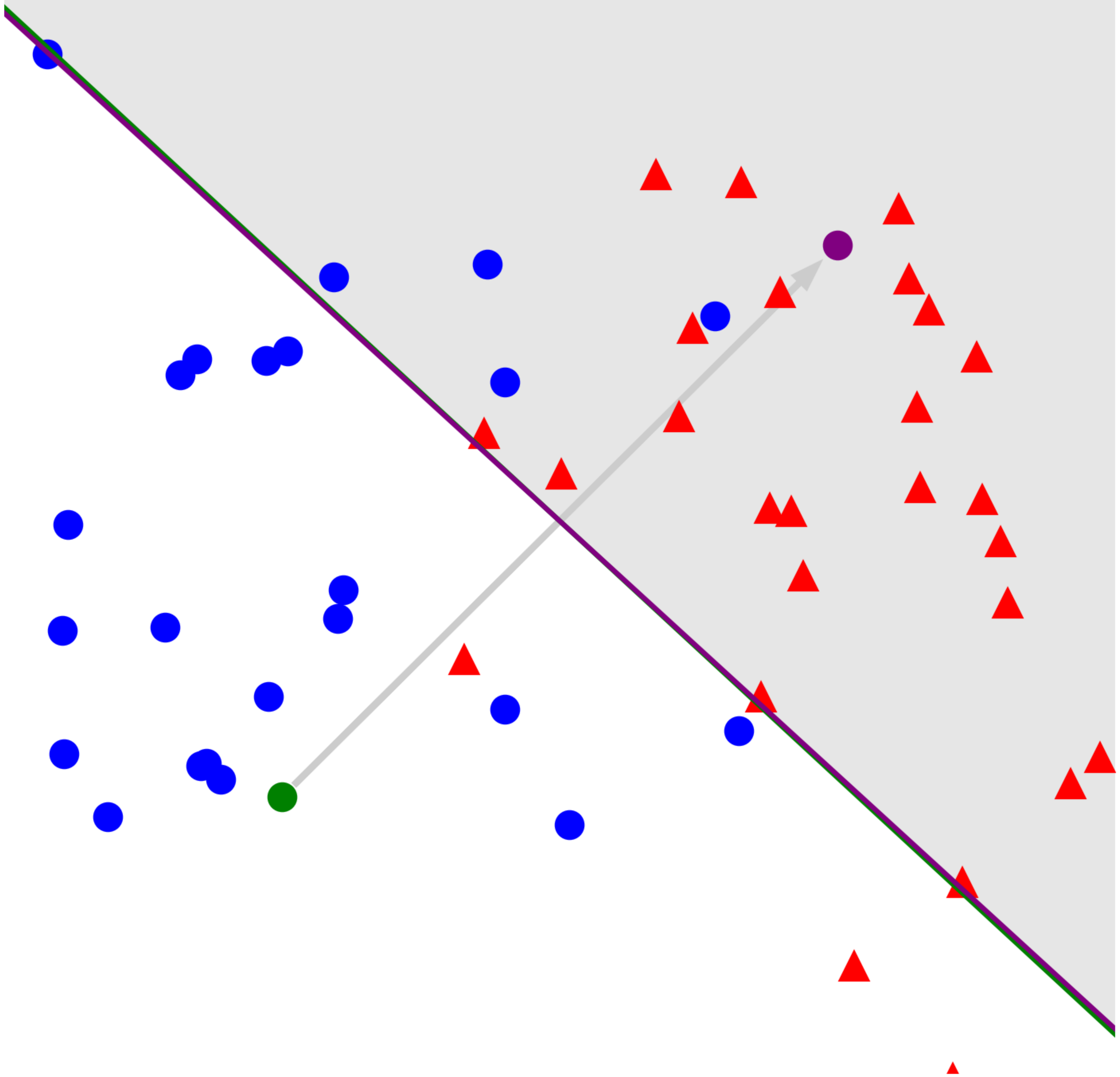
Step 12/30



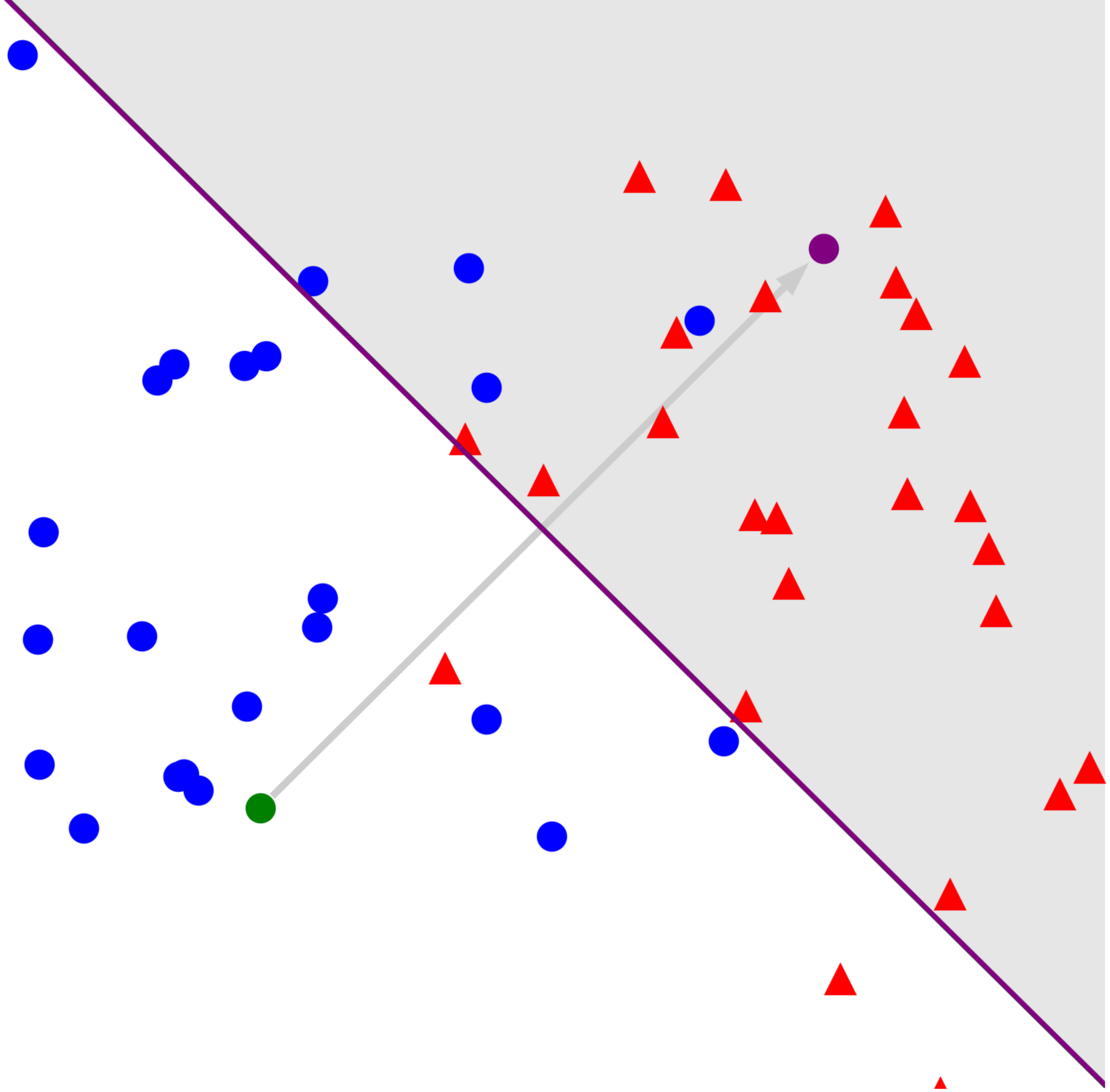
Step 13/30



Step 14/30

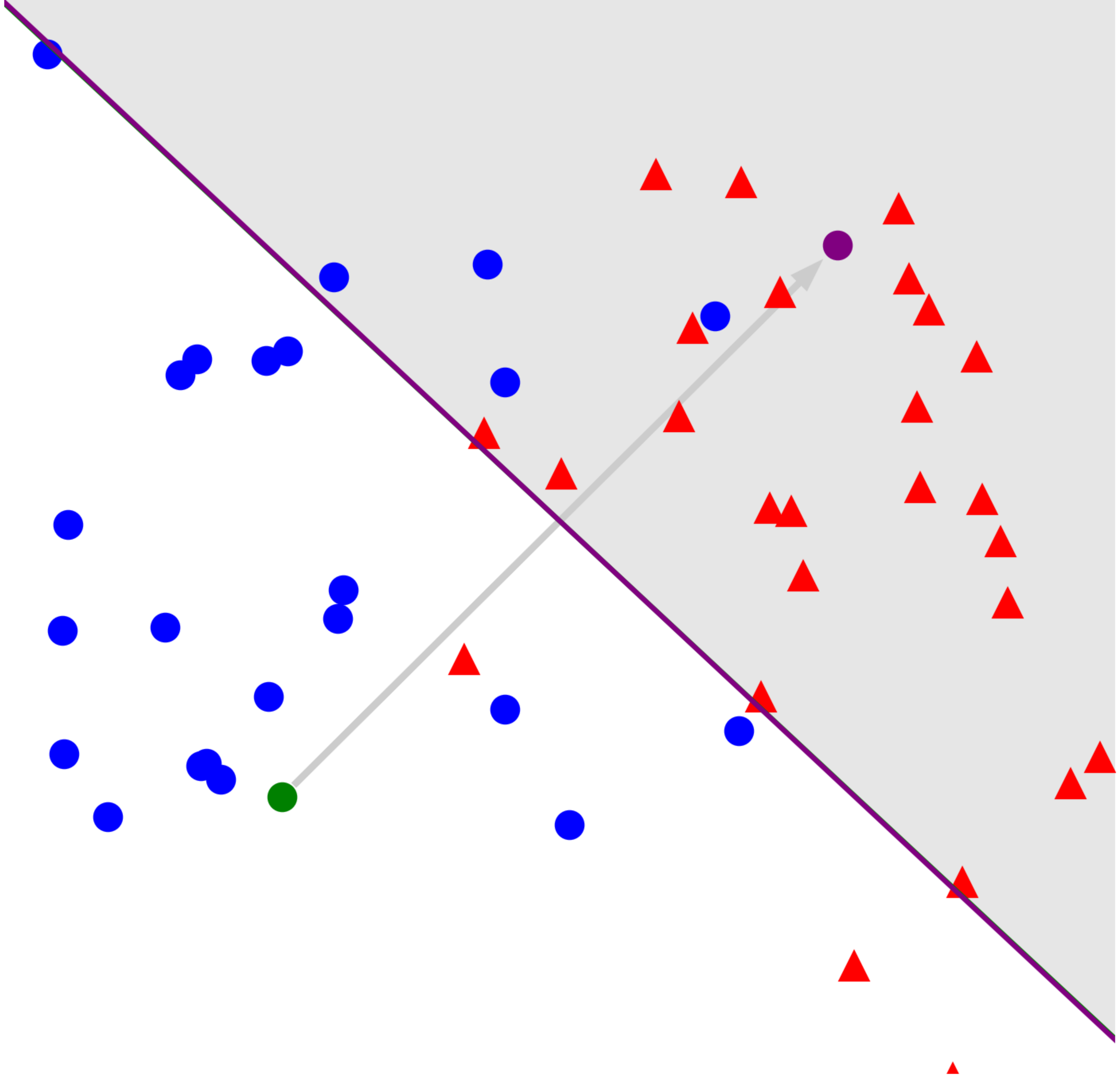


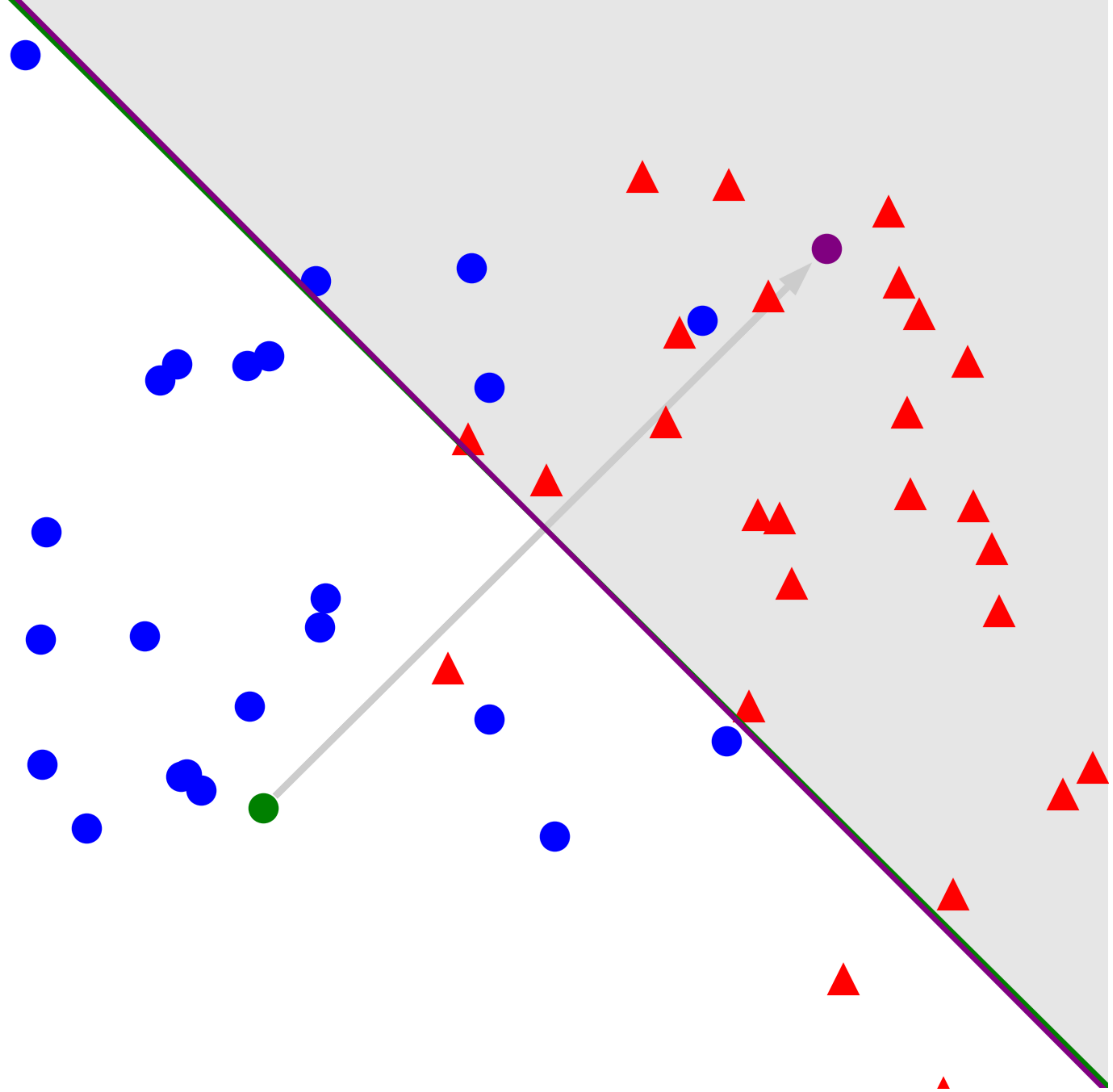
Step 15/30



Step 16/30

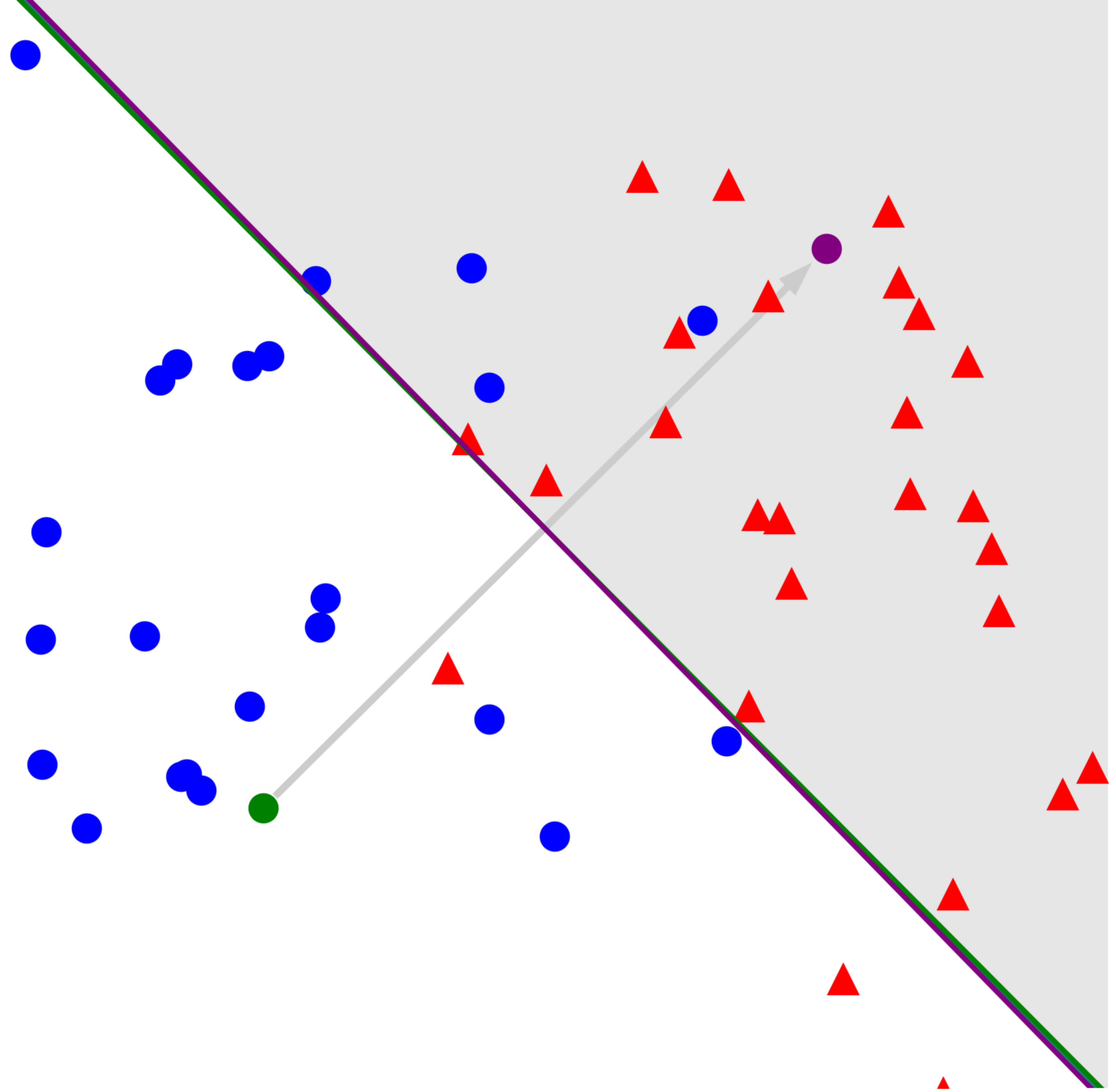
Step 17/30

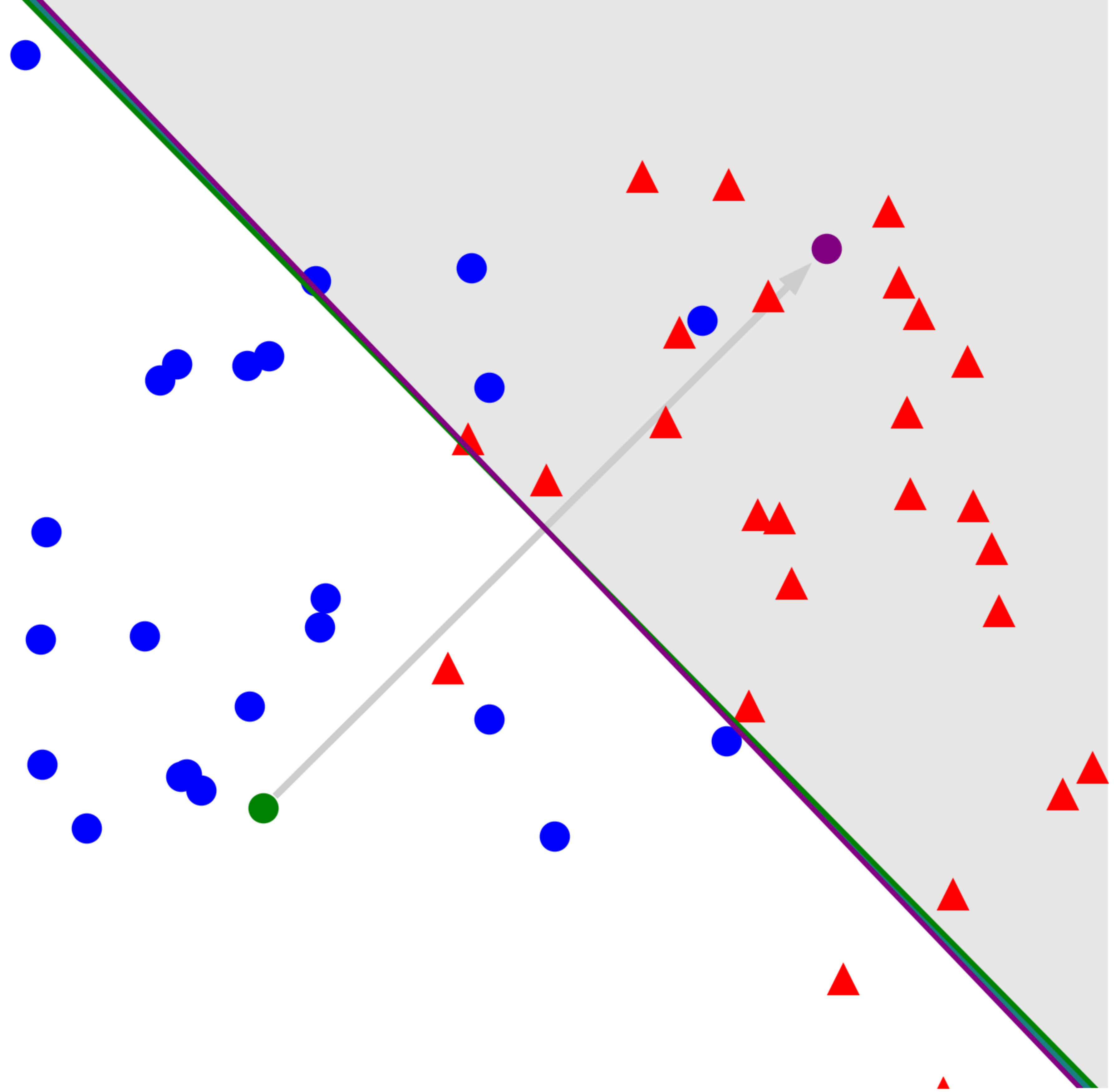




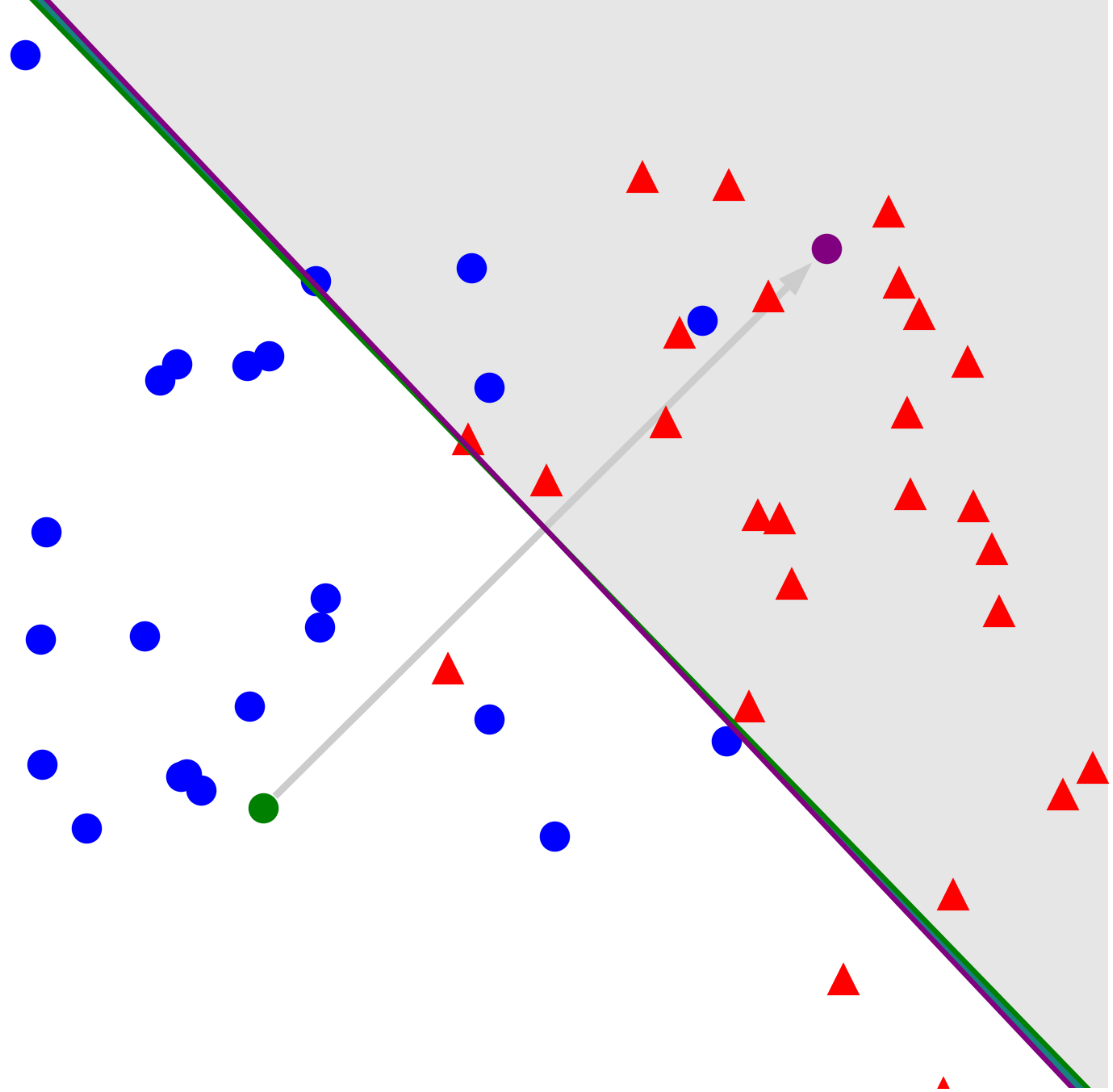
Step 18/30

Step 19/30

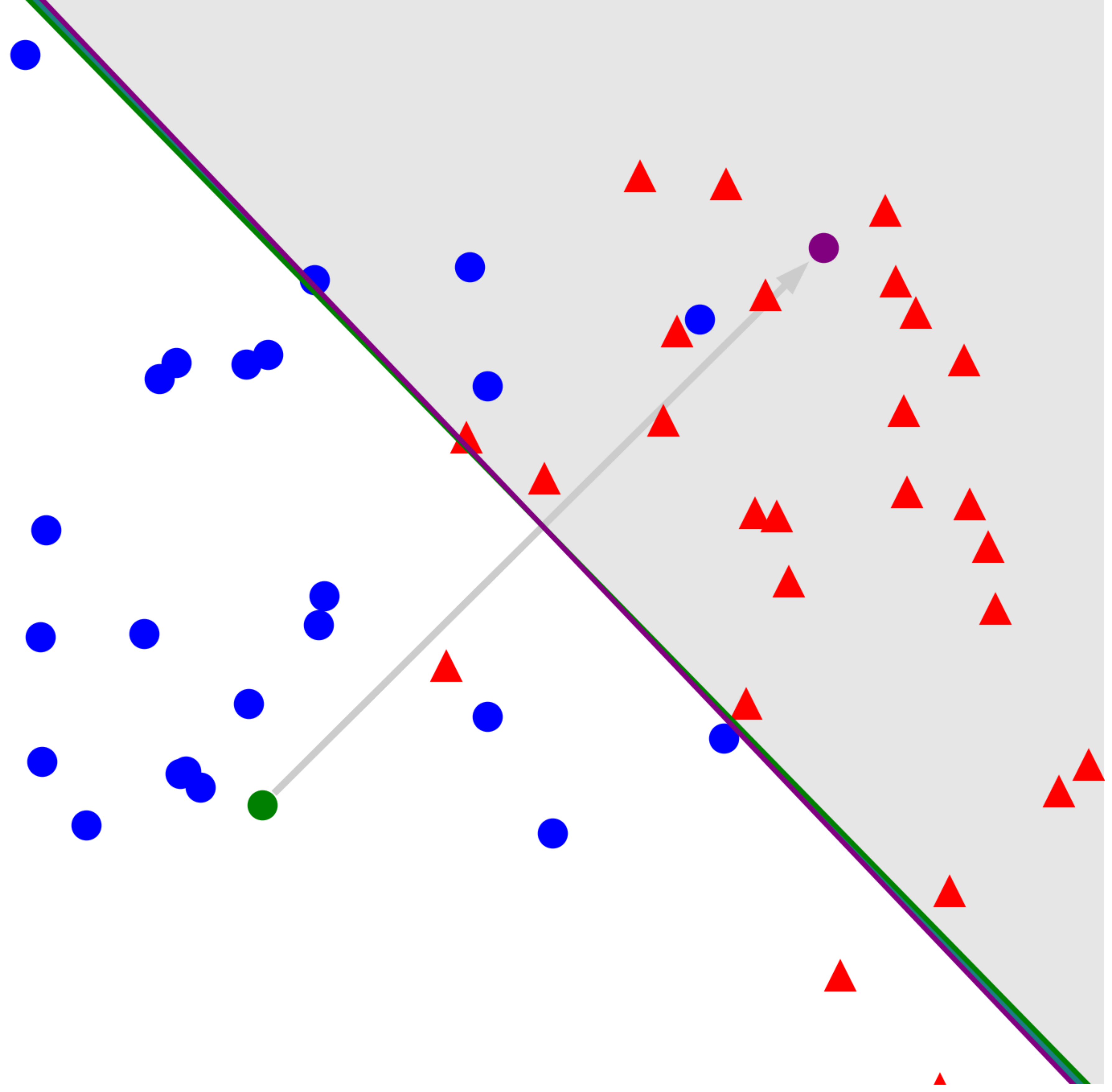




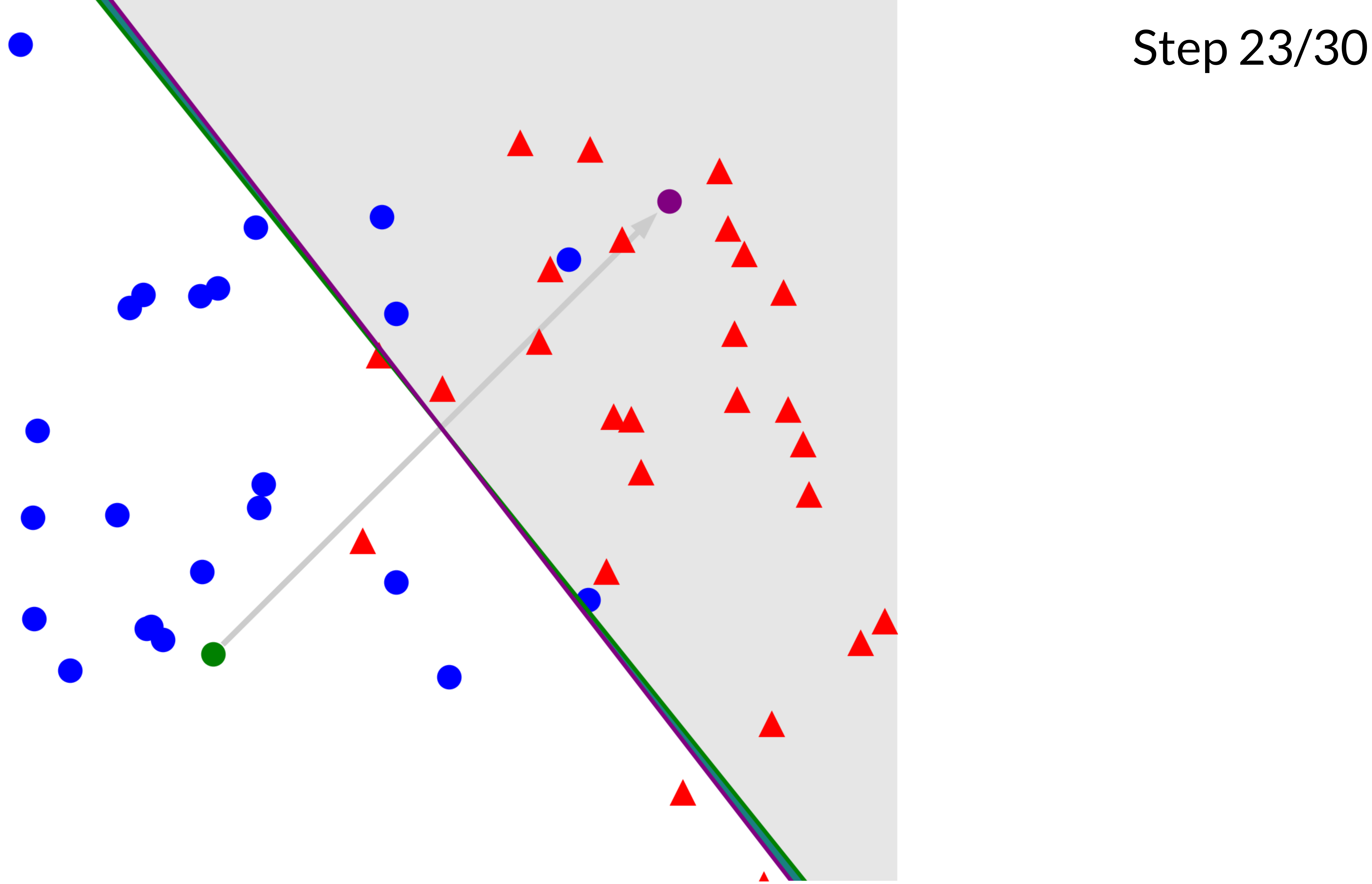
Step 20/30



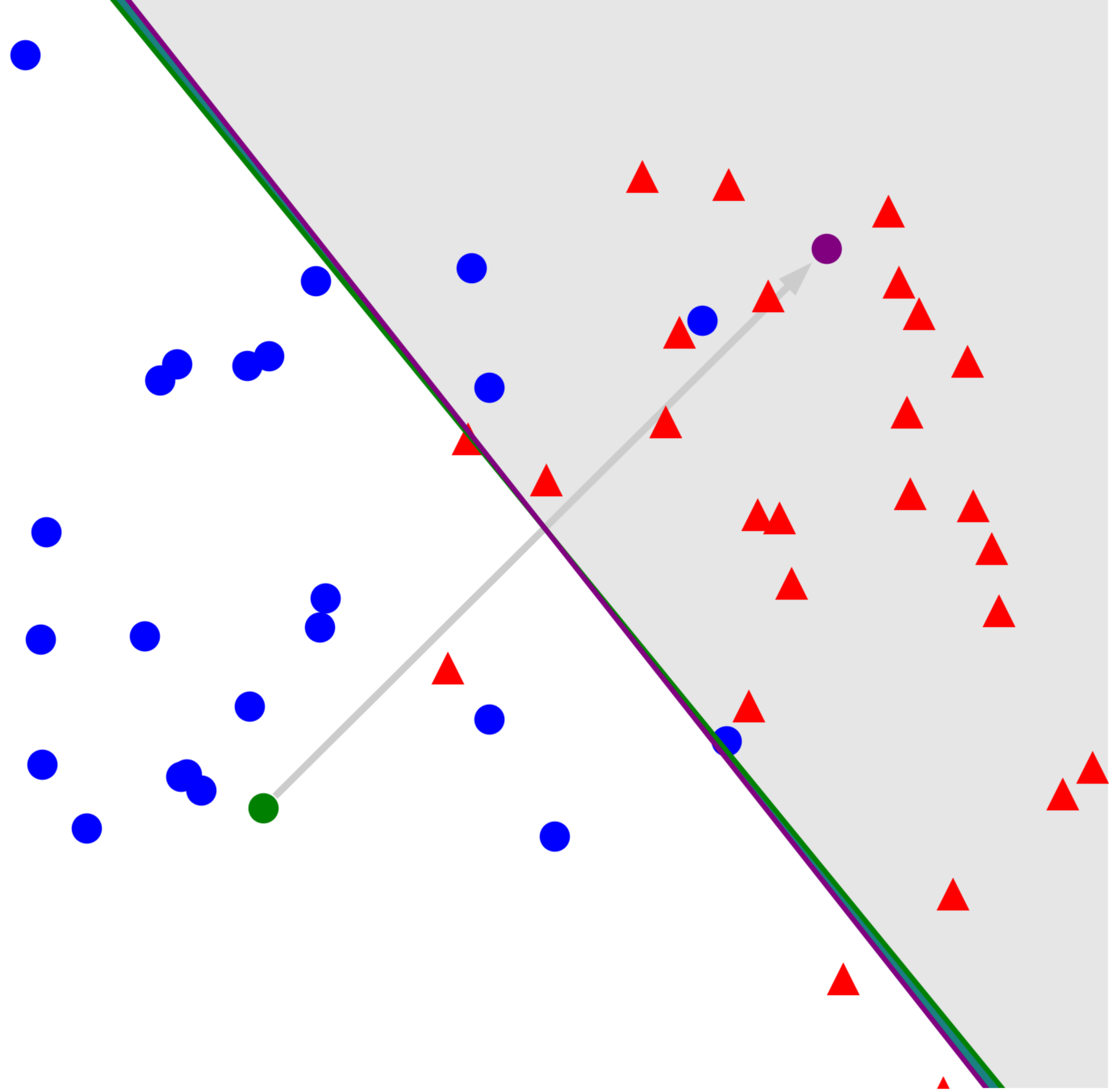
Step 21/30

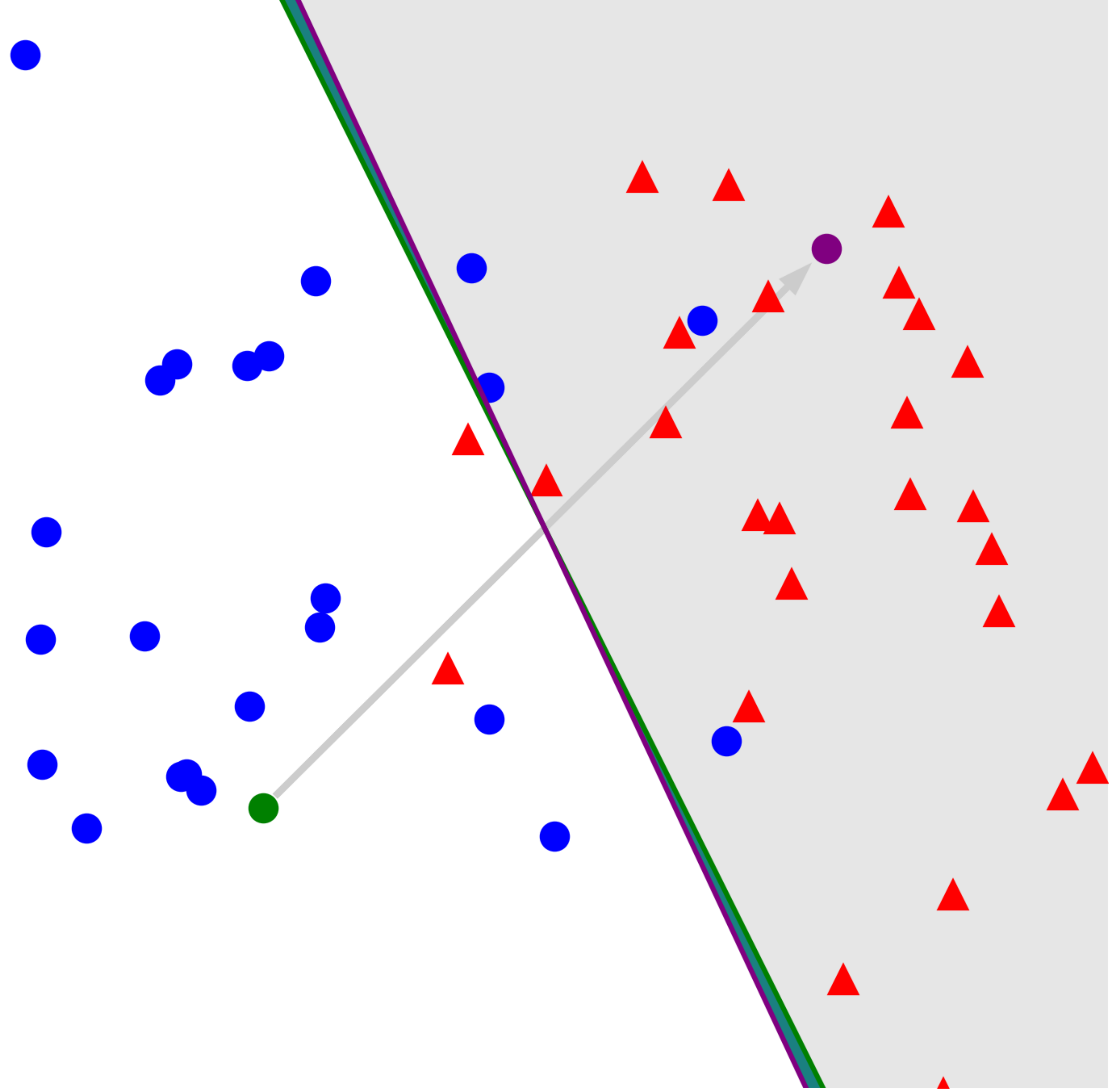


Step 22/30

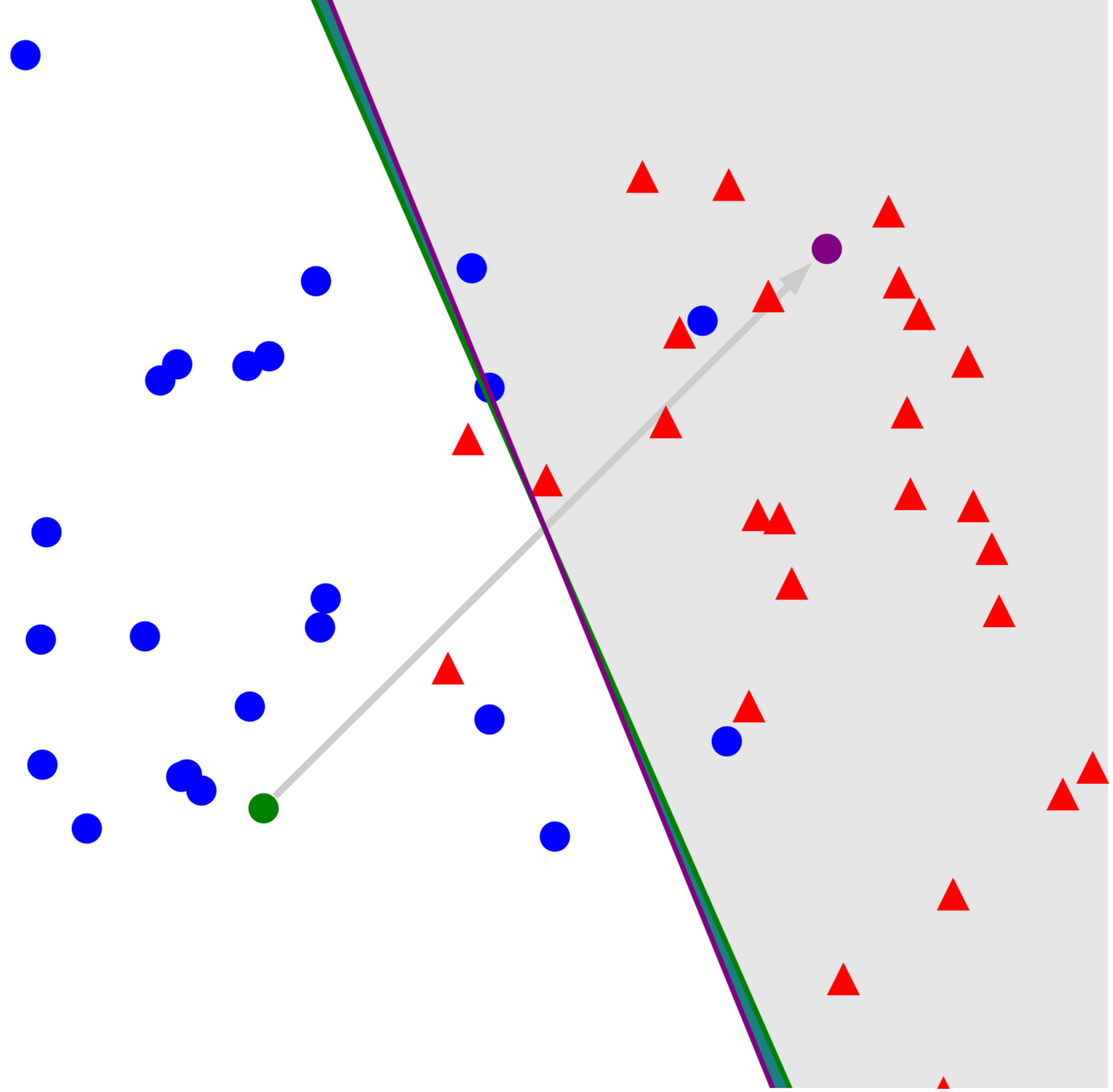

Step 23/30

Step 24/30

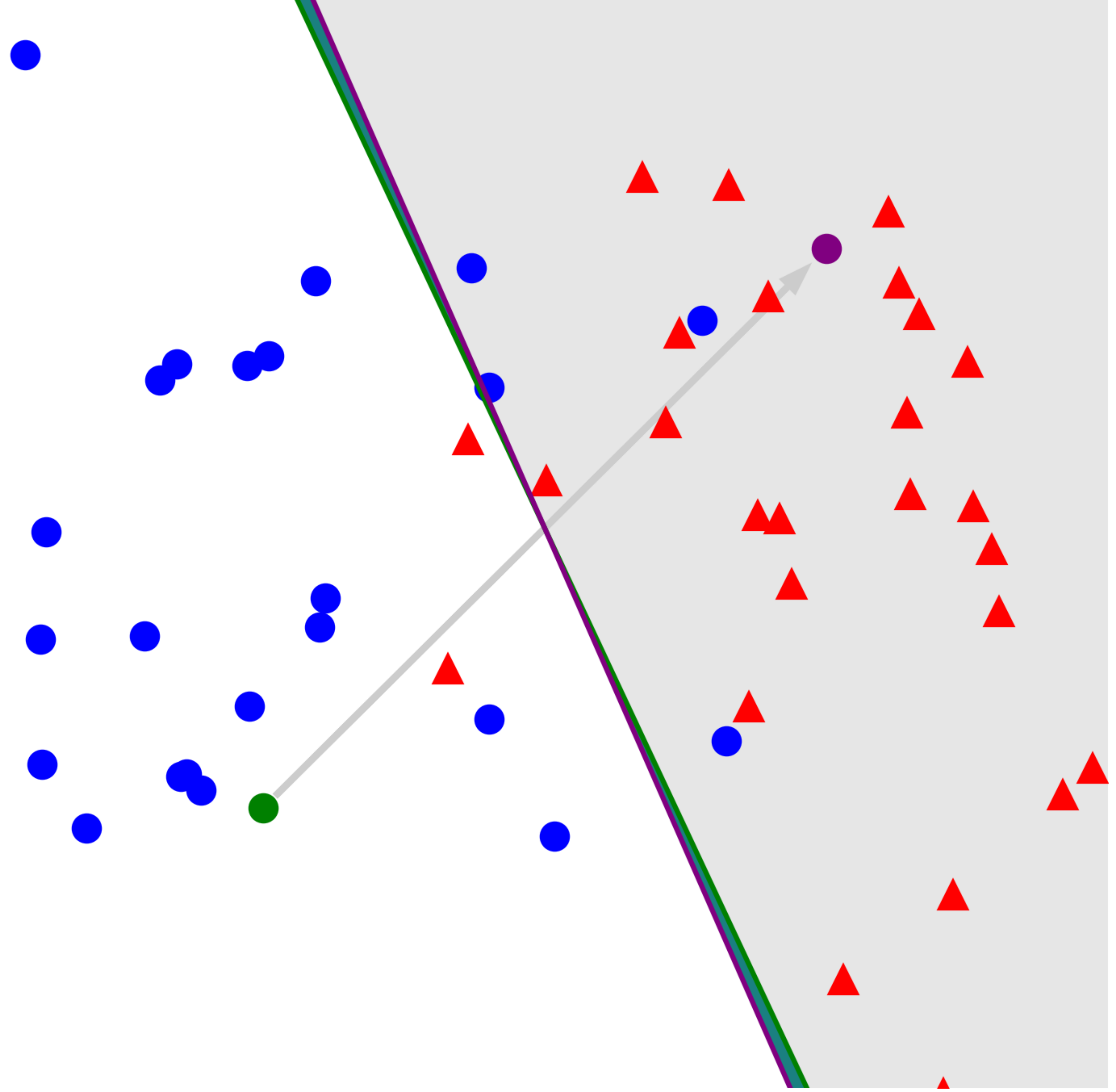




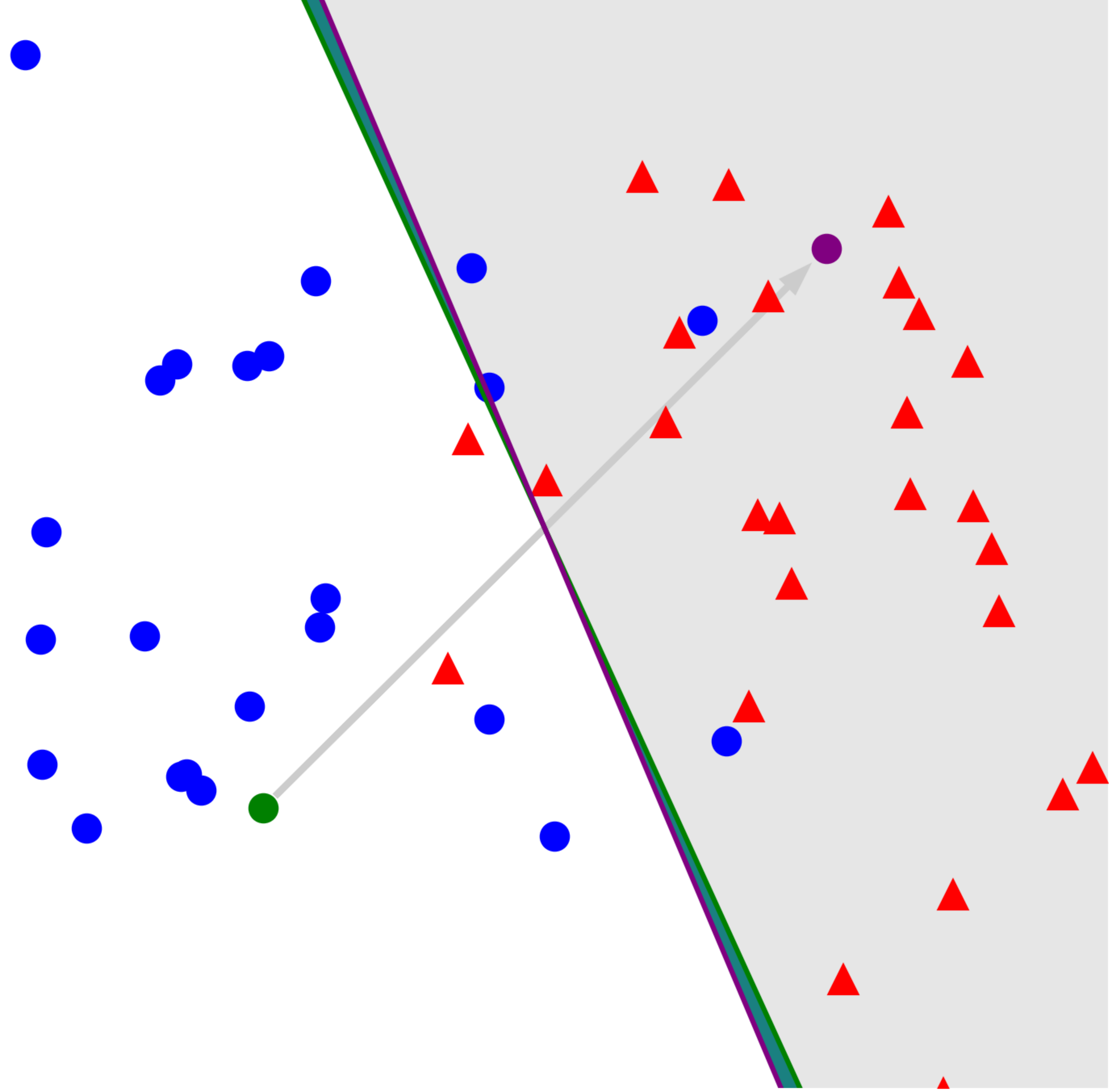
Step 25/30

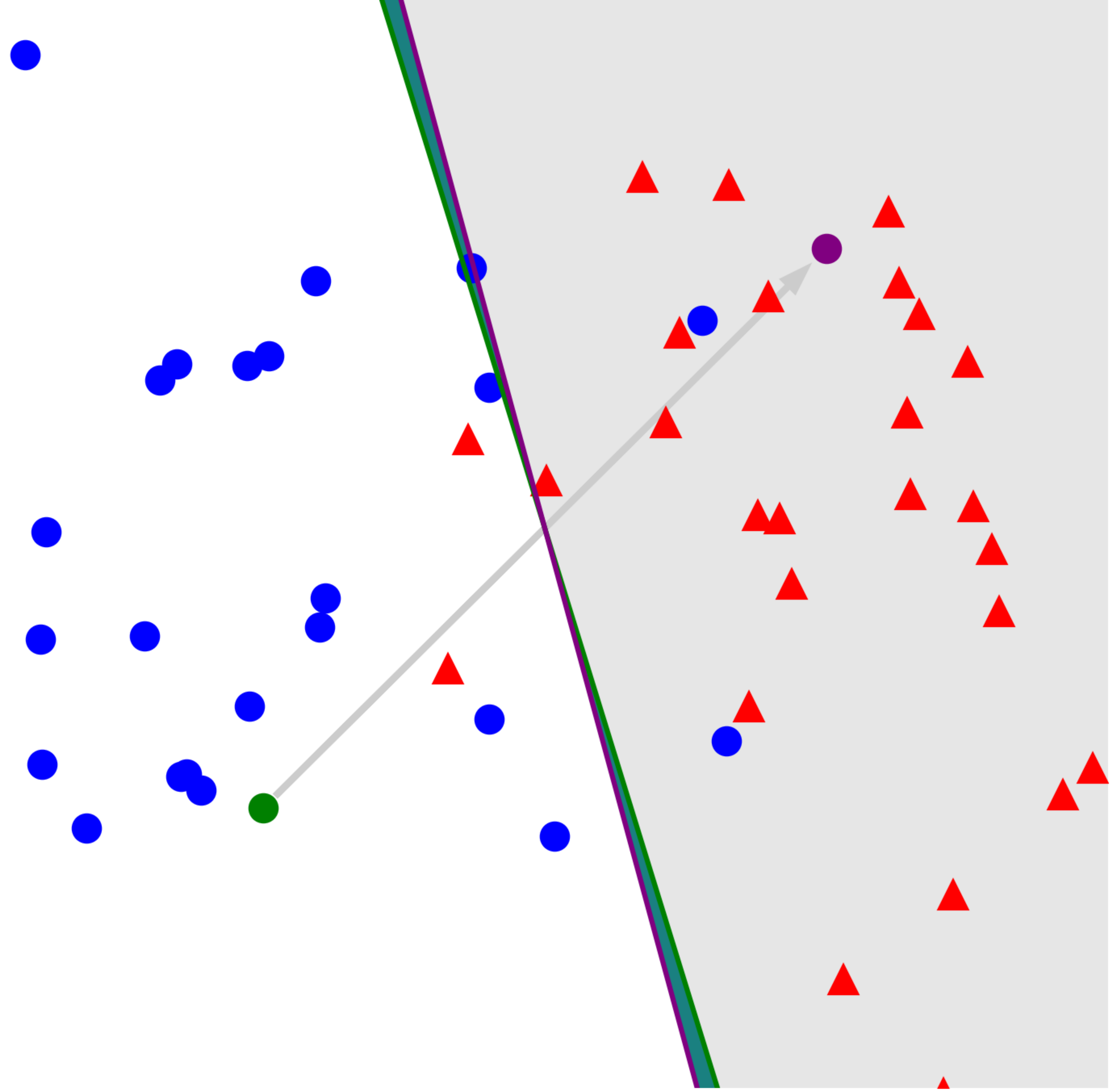


Step 27/30

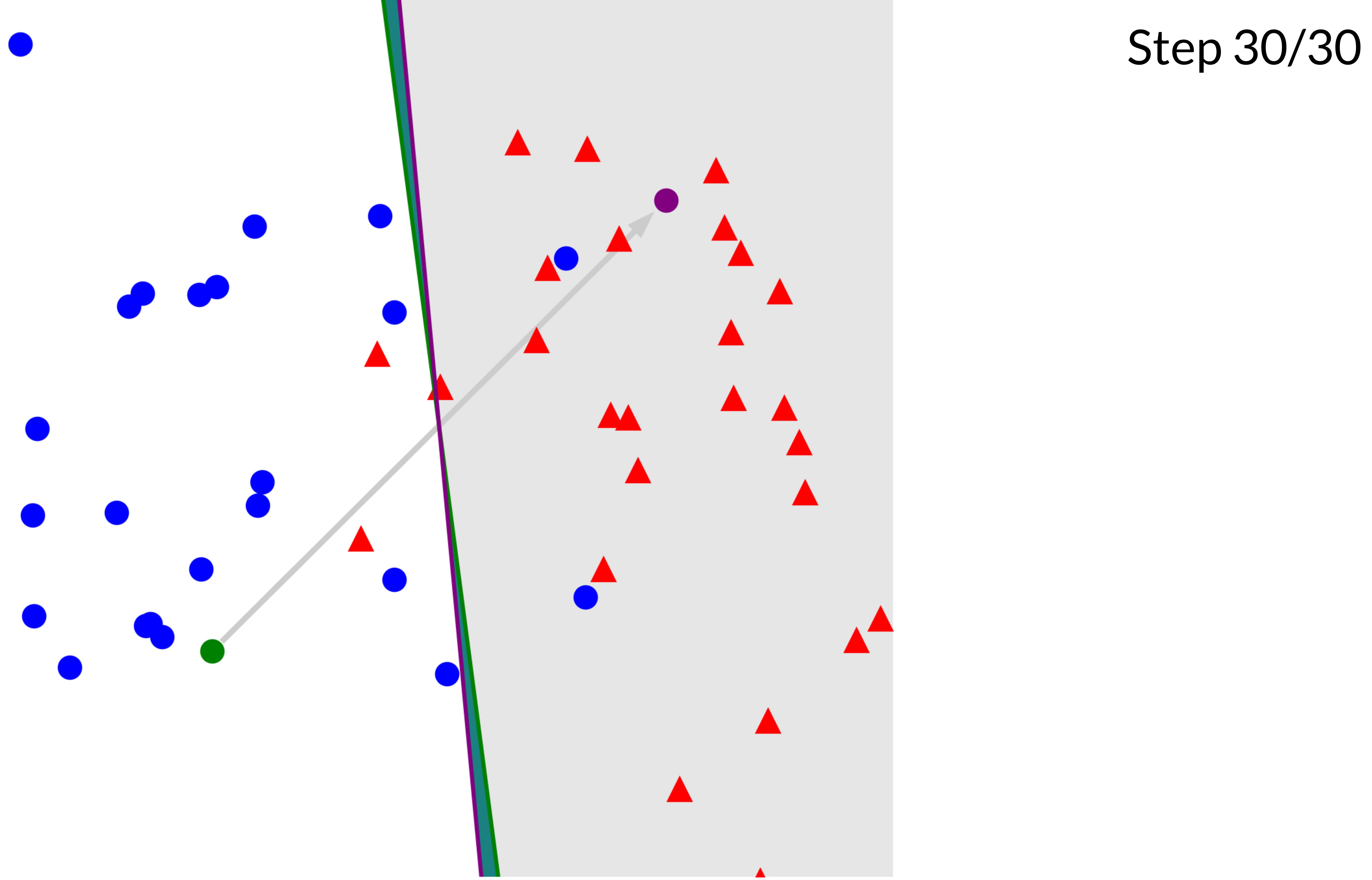


Step 28/30





Step 29/30


Step 30/30

The ghost sample

To introduce *stability* formally, we need two independent samples $S = (z_1, \ldots, z_n)$ and $S' = (z'_1, \ldots, z'_n)$

S' is called a **ghost sample** and serves an analytical purpose.

Introduce the hybrid sample $S^{(i)}$ as:

$$S^{(i)} = (z_1, \ \ldots, \ z_{i-1}, \ z'_i, \ z_{i+1}, \ \ldots \ , z_n)$$

Note that here the i-th example comes from S',
while all others come from S.

The **average stability** of an algorithm A:

$$\Delta(A) = \mathbb{E}_{S,S'} \left[\frac{1}{n} \sum_{i=1}^{n} \left(\ell(A(S), z_i') - \ell(A(S^{(i)}), z_i') \right) \right]$$

Expectations can be confusing. We can replace them by "max"

The **uniform stability** of an algorithm A is defined as

$$\Delta_{\sup}(A) = \max_{S,S'} \max_{i \in [n]} |\ell(A(S), z_i') - \ell(A(S^{(i)}, z_i')|$$

Note: $\Delta(A) \leq \Delta_{\sup}(A)$

Theorem

Average stability equals generalization gap.

$$\mathbb{E}[\epsilon_{\text{gen}}(A)] = \Delta(A)$$

Proof

$$\mathbb{E}[\epsilon_{\text{gen}}(A)] = \mathbb{E}\left[R(A(S)) - R_S(A(S))\right]$$

$$\mathbb{E}\left[R_S(A(S))\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\ell(A(S), z_i)\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\ell(A(S), z_i)]$$

$$\mathbb{E}[R(A(S))] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\ell(A(S), z_i')\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\ell(A(S), z_i')]$$

Since z_i and z_i' are identically distributed and independent of the other examples, we have

$$\mathbb{E}\ell(A(S), z_i) = \mathbb{E}\ell(A(S^{(i)}), z_i')\,.$$

Applying this identity to each term in above, we can see

$$\mathbb{E}[R(A(S)) - R_S(A(S))] = \Delta(A)$$

So, what learning algorithms are stable?

Theorem. Empirical risk minimization with any convex loss and ℓ_2-penalty is uniformly stable.

Click here for a proof.

Note: Generalization in non-convex learning is substantially more subtle

Conclusion

We contrasted risk and empirical risk

The difference between them equals a stability parameter

Stable algorithms can't overfit!

Interplay of robustness and generalization
is an active and fascinating research area.