11.03.2025

ERCİYES ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ A.B.D.
BİM-622 PEKİŞTİRMELİ ÖĞRENMEDE YENİ YAKLAŞIMLAR DERSİ ÖDEV-1

"Reinforcement Learning An Introduction"

Multi-armed Bandits Exercise Çözümleri

Ders Koordinatörü: Dr. Öğr. Üyesi Fehim KÖYLÜ

Dersi Alan Öğrenci: Ahmet Utku ELİK, 4010940078

Exercise 2.4

If the step-size parameters, a_n, are not constant, then the estimate Q_n is a weighted average of previously received rewards with a weighting different from that given by (2.6). What is the weighting on each prior reward for the general case, analogous to (2.6), in terms of the sequence of step-size parameters?

Eğer adım boyutu parametreleri a_n sabit değilse, o zaman Q_n tahmini, önceki alınan ödüllerin ağırlıklı bir ortalaması olur. Ancak, bu ağırlıklandırma, Denklem (2.6)'daki ağırlıklandırmadan farklı olacaktır. Genel durum için, her önceki ödülün ağırlıklandırması nasıl olur? Bu ağırlıklandırmayı, adım boyutu parametreleri dizisi cinsinden ifade ediniz.

Çözüm

Soruda bahsedilen Denklem (2.6) daşağı verilmiştir.

$$Q_{n+1} = (1 - \alpha)^n Q_1 + \sum_{i=1}^{n} \alpha (1 - \alpha)^{n-i} R_i \qquad (2.6)$$

Genel olarak kullanılan güncelleme kuralı Denklem (2.5)'de verilmektedir.

$$Q_{n+1} = Q_n + \alpha_n[R_n - Q_n] \qquad (2.5)$$

Bu denklemi açacak olursak aşağıdaki denklemi elde ederiz;

$$Q_{n+1} = (1 - \alpha_n)Q_n + \alpha_n R_n$$

$$Q_n = \sum_{i=1}^{n} w_i R_i$$

Bu denklemde yer alan ağırlıklar ise aşağıdaki denklem ile hesaplanabilir;

$$w_i = \prod_{j=i+1}^{n} (1 - \alpha_j), i < n, w_n = \alpha_n$$

Bu denklemden aşağıdaki sonuçları çıkartabiliriz;

1. Eğer a_n sabitse, önceki ödüllerin ağırlıkları üstel olarak azalır (denklem 2.6'ya benzer).

2. Eğer a_n değişkense, eski ödüller üzerindeki ağırlık değişebilir. Örneğin, büyük a_n değerleri daha yeni ödüllere daha fazla ağırlık verir.

Exercise 2.5

Design and conduct an experiment to demonstrate the difficulties that sample-average methods have for nonstationary problems. Use a modified version of the 10-armed testbed in which all the $q_*(a)$ start out equal and then take independent random walks (say by adding a normally distributed increment with mean zero and standard deviation 0.01 to all the $q_*(a)$ on each step). Prepare plots like Figure 2.2 for an action-value method using sample averages, incrementally computed, and another action-value method using a constant step-size parameter α=0.1. Use ε=0.1 and longer runs, say of 10,000 steps.

Örnek ortalama yöntemlerinin nonstationary (zamanla değişen) problemlerde yaşadığı zorlukları göstermek için bir deney tasarlayın ve uygulayın. 10-Armed Bandit test ortamının değiştirilmiş bir versiyonunu kullanın; burada tüm $q_*(a)$ başlangıçta eşit olmalı ve her adımda bağımsız rassal yürüyüşler gerçekleştirmelidir (her $q_*(a)$ değerine ortalaması sıfır ve standart sapması 0.01 olan normal dağılımlı bir artış ekleyerek).

Çözüm

Bu alıştırmada örnek ortalama (sample-average) ve sabit adım boyutu (constant step-size) yöntemlerinin nonstationary ortamlarda nasıl farklı performans gösterdiğini analiz etmemiz istenmektedir. Bu problemde, 10-Armed Bandit ortamındaki her kolun gerçek ödül değeri $q_*(a)$ başlangıçta eşit ama her adımda rastgele bir değişikliğe uğruyor. Yani, ödül fonksiyonu zamanla değişiyor (nonstationary environment).

Sample-Average Yöntemi;

Gelen tüm ödüllerin ortalamasını alarak aşağıdaki güncelleme formülünü kullanarak Q_n değerini günceller.

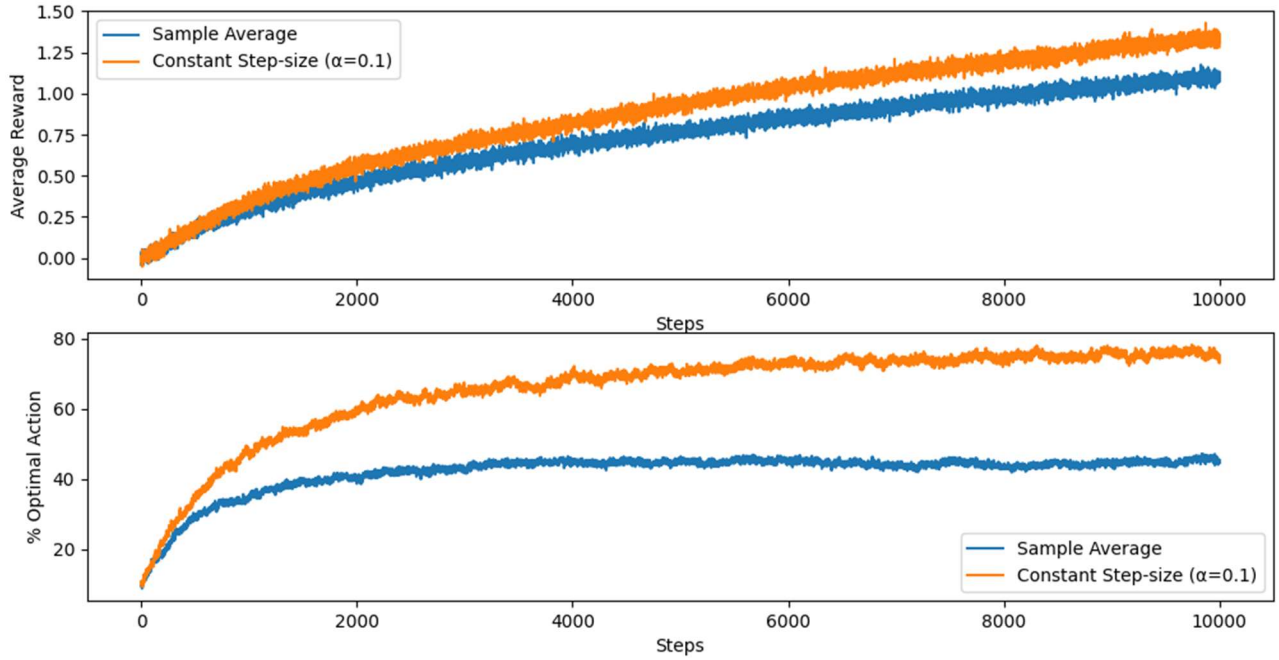
$$Q_n = Q_{n-1} + \frac{1}{n}(R_n - Q_{n-1})$$

Sabit Adım Boyutu (a = 0.1) Yöntemi;

Yeni gelen ödüllere daha fazla ağırlık verilmektedir. Güncelleme denklemi ise aşağıda verilmiştir.

$$Q_n = Q_{n-1} + \alpha(R_n - Q_{n-1})$$

İsterler dahilinde python programlama dili kullanılarak gerçekleştirilen deney sonucunda Şekil 1'de verilen grafik elde edilmiştir.



Şekil 1. Sample-Average ve Constant Step-Size Yöntemlerinin 10-armed Test Deneyindeki Ortalama Ödül – Optimal Eylem Yüzdesi Grafikleri

Elde edilen grafikler, ortalama ödül (average reward) ve optimal eylem yüzdesi (% optimal action) açısından iki yöntemin başarısını karşılaştırmaktadır. Bu grafiği inceleyecek olursak;

Grafiğin üst kısmında yer alan ortalama ödül eğrilerine bakıldığında, sabit adım boyutu yöntemi (a = 0.1, turuncu çizgi) zaman ilerledikçe daha yüksek ödüller elde etmektedir. Buna karşılık, örnek ortalama yöntemi (mavi çizgi) zamanla artmasına rağmen, sabit adım boyutuna sahip yönteme kıyasla daha düşük bir ortalama ödülde kalmaktadır. Bunun temel sebebi, örnek ortalama yönteminin eski ödülleri de hesaba katması nedeniyle zamanla değişen ödül ortamına hızlı adapte olamamasıdır. Sabit adım boyutu yöntemi ise daha güncel ödüllere daha fazla ağırlık verdiği için değişken ortama daha iyi uyum sağlayarak daha yüksek ödüller elde edebilmektedir.

Grafiğin alt kısmında yer alan optimal eylem yüzdesi eğrileri, hangi yöntemin daha sık optimal eylemi seçtiğini göstermektedir. Burada da sabit adım boyutu yönteminin optimal eylemi daha sık tercih ettiği gözlemlenmektedir. Özellikle, baştaki keşif aşamasından sonra turuncu eğri sürekli olarak mavi eğrinin üstünde seyretmekte ve optimal eylem yüzdesinin daha yüksek olduğu görülmektedir. Bunun sebebi, sabit adım boyutu yönteminin ödülleri hızlı güncellemesi ve en iyi eylemi daha çabuk öğrenebilmesidir. Örnek ortalama yöntemi ise geçmişe bağlı kalarak eski ödüllerin etkisini fazla taşıdığı için optimal eyleme ulaşma konusunda daha yavaş hareket etmektedir.

Sonuç olarak nonstationary ortamlar için sabit adım boyutu yöntemi (α=0.1) daha iyi bir seçimdir. Örnek ortalama yöntemi, istatistiksel olarak sağlam bir yaklaşım olsa da dinamik değişikliklere karşı daha yavaş tepki verir. Bu nedenle, nonstationary bandit problemlerinde sabit adım boyutlu öğrenme yöntemi kullanılarak daha hızlı adaptasyon sağlanabilir ve daha yüksek ortalama ödül elde edilebilir.

Projenin Github Kaynak Kodu: https://github.com/Rexoes/BIM622

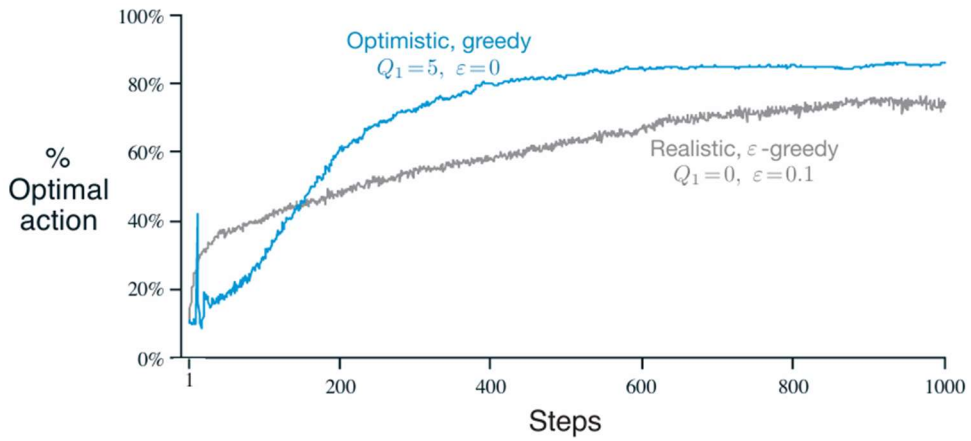
Exercise 2.6

Mysterious Spikes - The results shown in Figure 2.3 should be quite reliable because they are averages over 2000 individual, randomly chosen 10-armed bandit tasks. Why, then, are there oscillations and spikes in the early part of the curve for the optimistic method? In other words, what might make this method perform particularly better or worse, on average, on particular early steps?

Gizemli Zirveler - Şekil 2.3'te gösterilen sonuçlar oldukça güvenilir olmalıdır, çünkü bunlar rastgele seçilen 10-kollu bandit görevleri üzerinde 2000 bağımsız denemenin ortalamalarıdır. Peki o halde, iyimser yöntem için eğrinin ilk kısmında neden dalgalanmalar (oscillations) ve ani yükselişler (spikes) görülmektedir? Başka bir deyişle, bu yöntem belirli erken adımlarda neden ortalama olarak özellikle daha iyi veya daha kötü performans gösterebilir?

Çözüm

Soruda bahsi geçen ve aşağıda verilen Şekil 2.3 Optimistic, Greedy ile Realistic, ε Greedy yöntemlerinin karşılaştırmasını göstermektedir.



Şekil 2.3. Optimistic, Greedy - Realistic, ε Greedy Karşılaştırması

Optimistic yöntem (mavi çizgi) için erken adımlarda dalgalanmalar görülmektedir bunun sebebi, ajan başlangıçta tüm eylemlerin iyi olduğunu düşündüğü için bir süre boyunca yanlış eylemlere bağlı kalabilir ancak ödüller tahmin edilenden düşük olduğunda, ajan hızla diğer eylemleri denemeye başlar ve keşif artar. Bazı adımlarda ani performans artışları yaşanır, çünkü ajan bir noktada en iyi kolları keşfeder ve daha sık kullanmaya başlar.

Optimistic yöntem bazı erken adımlarda daha kötü performans göstermesinin nedeni eğer iyimser başlangıç değerleri yanlışsa (yani en iyi ödüle sahip olmayan bir eylem en yüksek tahminle başlatılmışsa), ajan yanlış seçimler yapar ve ilk birkaç adımda gereksiz yere kötü ödüller alabilir. Bu yöntemin devam eden adımlarda daha iyi performans göstermesinin nedeni ise İyimser başlangıç değerleri, ajanı keşif yapmaya teşvik eder ve uzun vadede en iyi eylemi bulmasını kolaylaştırmaktadır.

Sonuç olarak başlangıç değerlerinin öğrenme sürecindeki önemi anlaşılmaktadır. Optimistic başlangıç değerleri, ajanı erken keşif yapmaya zorlayarak uzun vadede daha iyi performans sağlamasına yardımcı olmaktadır ancak bu yöntem erken aşamalarda istikrarsızlığa ve bazı adımlarda ani zirvelere (spikes) neden olmaktadır.

Optimistic başlangıç değerleri, ilk adımlarda dalgalanma ve düşük performansa, ancak uzun vadede daha iyi adaptasyona neden olmaktadır. Gerçekçi ε Greedy yöntemi ise başlangıçta daha stabil öğrenmeye, ancak uzun vadede optimal eylemi bulmada yavaş kalmaktadır.

Exercise 2.7

Unbiased Constant-Step-Size Trick - In most of this chapter, we have used sample averages to estimate action values because sample averages do not produce the initial bias that constant step sizes do (see the analysis leading to (2.6)). However, sample averages are not a completely satisfactory solution because they may perform poorly on nonstationary problems. Is it possible to avoid the bias of constant step sizes while retaining their advantages on nonstationary problems? One way is to use a step size of:

$$\beta_n \doteq \frac{\alpha}{\bar{o}_n} \tag{2.8}$$

to process the nth reward for a particular action, where a > 0 is a conventional constant step size, and $\bar{o}_n$ s a trace of one that starts at 0:

$$\bar{o}_n \doteq \bar{o}_{n-1} + \alpha\,(1 - \bar{o}_{n-1}), for\ n \geq 0, with\ \bar{o}_n \doteq 0. \tag{2.9}$$

Carry out an analysis like that in (2.6) to show that Q_n is an exponential recency-weighted average without initial bias.

Önyargısız Sabit Adım Boyutu Hilesi - Bu bölümün büyük bir kısmında, eylem değerlerini tahmin etmek için örnek ortalamaları kullandık çünkü örnek ortalamalar sabit adım boyutlarının oluşturduğu başlangıç önyargısını üretmez (bkz. Denklem (2.6)). Ancak, örnek ortalamalar nonstationary (zamanla değişen) problemler için her zaman iyi bir çözüm sunmaz. Peki, sabit adım boyutlarının önyargısını ortadan kaldırırken aynı zamanda nonstationary problemlerdeki avantajlarını korumak mümkün müdür? Bunun bir yolu, aşağıdaki adım boyutunu kullanmaktır:

$$\beta_n \doteq \frac{\alpha}{\bar{o}_n} \tag{2.8}$$

Burada a > 0 klasik bir sabit adım boyutudur ve $\bar{o}_n$ başlangıçta 0 olan bir iz değişkenidir (trace variable):

$$\bar{o}_n \doteq \bar{o}_{n-1} + \alpha\,(1 - \bar{o}_{n-1}), for\ n \geq 0, with\ \bar{o}_n \doteq 0. \tag{2.9}$$

Denklem (2.6) benzeri bir analiz yaparak, Q_n 'nin başlangıç önyargısı olmadan üstel olarak güncellenmiş bir ağırlıklı ortalama olduğunu gösteriniz.

Çözüm

Denklem (2.6)'da verilen standart sabit adım boyutlu güncelleme denklemi aşağıda verilmiştir;

$$Q_{n+1} = (1 - \alpha)^n Q_1 + \sum_{i=1}^{n} \alpha (1 - \alpha)^{n-i} R_i \quad (2.6)$$

Denklemde yer alan $(1 - \alpha)^n Q_1$ terimi, başlangıç önyargısını gösterir ve bu önyargı n sonsuz olduğunda sıfıra yaklaşsa da erken adımlarda önemli bir etkiye sahiptir.

Soruda önerilen yöntem, adım boyutunu zamanla değiştirerek başlangıç önyargısını azaltmaktır. Bunu sağlamak için adım boyutu Denklem (2.8, 2.9) kullanılarak değiştirilir. Denklemde yer alan $\bar{o}_n$, 1'e yakınsayacak şekilde güncellenen bir değişkendir. Bu iz değişkeni, sabit bir a yerine daha kontrollü bir ağırlıklandırma sağlar, böylece ilk adımlarda büyük bir ağırlık verilerek öğrenme süreci dengelenir.

Yeni yöntemde, adım boyutu Denklem (2.8)'de verilen β_n olduğundan, güncelleme kuralı şu şekilde değişir;

$$Q_{n+1} = Q_n + \beta_n[R_n - Q_n] \quad (2.5)$$

Bunu tekrar tekrar açarak Denklem (2.6)'ya benzer şekilde genişletirsek;

$$Q_n = \sum_{i=1}^{n} w_i R_i$$

denklemini elde ederiz. Burada yer alan ağırlık katsayıları ise aşağıdaki denklem ile hesaplanır;

$$w_i = \beta_i \prod_{j=i+1}^{n} (1 - B_j), \quad i < n, \quad w_n = \beta_n$$

Bu analiz, Q_n 'nin başlangıç önyargısı olmadan hesaplandığını ve yine de üstel olarak ağırlıklı bir ortalama ürettiğini göstermektedir. Bu yöntem, eski verilerin etkisini zamanla azaltırken başlangıç önyargısını da ortadan kaldırır. Bu sayede, hem sabit adım boyutunun avantajlarını koruyarak hızlı öğrenme sağlanır hem de erken aşamalardaki sapmalar engellenir.

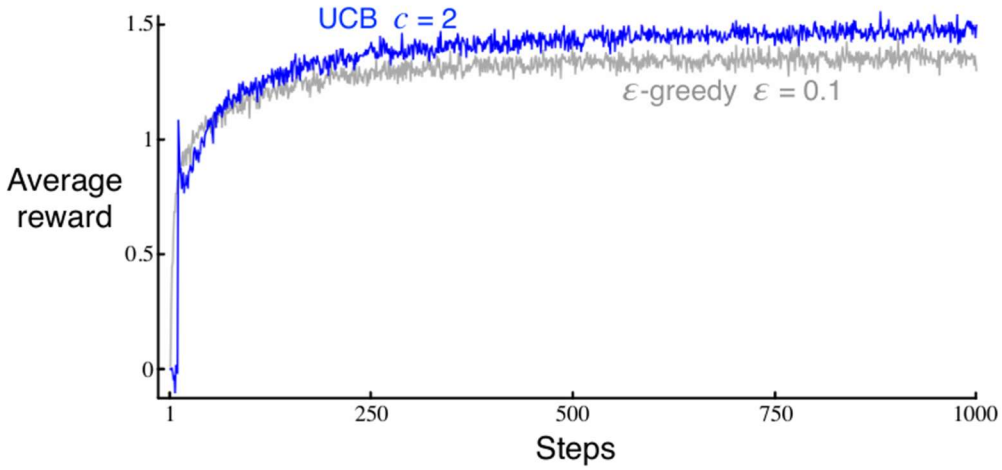
Exercise 2.8

UCB Spikes - In Figure 2.4, the UCB algorithm shows a distinct spike in performance on the 11th step. Why is this? Note that for your answer to be fully satisfactory, it must explain both why the reward increases on the 11th step and why it decreases on the subsequent steps. Hint: if c = 1 , then the spike is less prominent.

UCB'deki Zirve (Spikes) - Şekil 2.4'te görüldüğü gibi, UCB algoritması 11. adımda belirgin bir performans zirvesi gösteriyor. Bunun sebebi nedir? Cevabınızın tam olması için hem 11. adımda ödülün neden arttığını hem de takip eden adımlarda neden azaldığını açıklamanız gerekmektedir. İpucu: Eğer c = 1 olarak seçilirse, bu zirve daha az belirgin olur.

Çözüm

Soruda bahsedilen Şekil 2.4 UCB eylem seçiminin 10-armed testindeki ortalama performans grafiği aşağıda verilmiştir.



Şekil 2.4. UCB Eylem Seçiminin 10-armed Testindeki Ortalama Performans Grafiği

Öncelikle UCB algoritmasının çalışma prensibini anlamak gerekmektedir. UCB algoritması, keşif (exploration) ve sömürü (exploitation) dengesini matematiksel olarak kontrol eden bir algoritmadır. Seçilecek eylem, Denklem (2.10) ile belirlenmektedir;

$$A_t = argmax_a \left[Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right] \tag{2.10}$$

- $Q_t(a)$: Eylem a için ortalama ödül tahmini.
- Nt(a) : Eylem a'nın kaç kez seçildiği.
- c : Keşif (exploration) faktörünü belirleyen hiperparametre.

Şekil 2.4'te görülen 11. Adımda UCB eylem seçimindeki ani performans artışının sebebini şöyle açıklayabiliriz.UCB algoritmasında başlangıçta tüm kolların keşfedilmesi gerekir, bu nedenle UCB tüm kolları en az bir kez dener.İlk 10 adımda, her kol en az bir kez seçildiğinden Nt(a) değerleri eşitlenir. 11. Adımda ise UCB algoritması keşif bileşeni sayesinde şimdiye kadar en az denenmiş ve en yüksek belirsizliğe sahip olan kolu seçer. Eğer bu kol gerçekten yüksek bir ödüle sahipse, bu adımda ortalama ödülde ani bir sıçrama gözlemlenir.

UCB algoritmasının devam eden adımlarda performansının düşme sebebini açıklayacak olursak. UCB keşif yapmayı bırakarak daha önce iyi performans göstermiş kolları sömürmeye başlar. Eğer 11. adımda yüksek bir ödül alınmışsa, sonraki birkaç adımda diğer kolların keşfi devam ettiğinden ödül biraz düşebilir. Keşif oranı azaldıkça, ödüller stabil hale gelir ve zirveden düşüş yaşanır. Fakat sonuç olarak Şekil 2.4'de görüldüğü gibi, UCB zamanla dengeye oturarak en iyi eylemleri seçmeye başlar.

Soruda ipucu olarak verilen UCB algoritmasında seçilecek olan eylem denkleminde yer alan Keşif (exploration) faktörünü belirleyen hiperparametre olan c değerinin etkisini inceleyecek olursak;

- **Büyük c değeri (c=2):** UCB daha fazla keşif yapar ve düşük denenmiş kolları daha sık seçer. Dolayısı ile erken adımlarda daha fazla dalgalanmalara (spikes) neden olur.

- **Küçük c değeri (c=1):** UCB daha az agresif keşif yapar, yani erken aşamalardaki dalgalanmalar azalır.

Sonuç olarak 11. adımda ani performans artışı görülmesinin sebebi, UCB algoritmasının en az denenmiş kolu seçerek büyük bir keşif yapmasıdır. Takip eden adımlarda ödülün düşme sebebi ise algoritmanın tekrar keşif ve sömürü dengesine oturması ve önceki yüksek ödüllerin geçici olmasıdır. Kontrol parametresi olan c değeri artırıldığında, erken aşamalardaki dalgalanmalar daha belirgin hale gelir, çünkü algoritma daha fazla keşif yapmaya zorlanır. Daha düşük c değerlerinde ise erken aşamalardaki dalgalanmaları azaltabilir ancak uzun vadede daha yavaş öğrenmeye neden olabilir.