

# “Take A Shortcut Back: Mitigating the Gradient Vanishing for Training Spiking Neural Networks” Makalesinin Derinlemesine İncelemesi

## Giriş (Introduction)

**Spiking Sinir Ağları (SNN):** Spiking sinir ağları, biyolojik sinir sistemlerinden ilham alan özel bir yapay sinir ağı türüdür. Bu ağlar **ikili (0/1) “diken” sinyallerini** kullanarak bilgi iletir; bir nöronun **membran potansiyeli belirli bir eşiği aştığında 1 ile temsil edilen bir spike (diken) üretir**, aksi halde 0 olur <sup>1</sup>. Bu sayede ağırlıklar ve aktivasyonlar arasındaki çarpımlar yerine basit toplamalar kullanılır, yani hesaplama yükü azalır <sup>2</sup>. Dahası, SNN’ler **olay güdümlü (event-driven)** çalışarak yalnızca bir spike oluştuğunda hesaplama yapar, spike olmadığında işlemci beklemede kalarak enerji tasarrufu sağlar <sup>3</sup>. Bu paradigma sayesinde SNN’ler, eşdeğer yapay sinir ağlarına (ANN) kıyasla **büyüklikler mertebesinde daha az enerji** tüketebilir <sup>4</sup>. SNN’lerin sağladığı yüksek enerji verimliliği, onları **nesne tanıma, nesne algılama, poz takibi** gibi çeşitli alanlarda ve özellikle **nöromorfik donanımlarda** tercih edilen bir yöntem haline getirmiştir <sup>5</sup>.

**Eğitim Zorluğu – Gradyan Kaybolması:** SNN’lerin tüm bu avantajlarına karşın, **doğrudan eğitilmeleri ciddi bir zorluk** barındırır. Bunun temel sebebi, bir SNN’deki **spike üretme fonksiyonunun türevinin tanımsız olmasıdır**. Yani, normal geri yayılım (backpropagation) algoritması SNN’lere doğrudan uygulanamaz <sup>6</sup> <sup>7</sup>. Bu sorunu aşmak için literatürde, **“ikame (surrogate) gradyan” yöntemleri** geliştirilmiştir <sup>8</sup>. İkame gradyan yaklaşımında, ileri aktarım sırasında ağ normal şekilde spike üretirken, geri yayılım sırasında türev alınamayan spike fonksiyonu yerine türevi tanımlı *yaklaşık* bir fonksiyon kullanılır <sup>9</sup>. Örneğin, literatürde **kesik parabolik, sigmoid, tanh-benzeri veya dikdörtgen** gibi farklı ikame fonksiyonlar önerilmiştir <sup>10</sup>. Bu sayede SNN’ler, tıpkı sürekli değerli ağlar gibi, gradient tabanlı optimizasyonla eğitilebilmektedir.

Ancak, önceki çalışmalar bu yaklaşımları kullanırken **içsel bir sorunu göz ardı etmişlerdir: gradyan kaybolması** <sup>11</sup>. İkame fonksiyonlarının tümü, orijinal spike fonksiyonu gibi **sınırlı (bounded)** çıktı verdiğinden türevleri de belirli aralıklar dışında **0’a çok yakındır** <sup>12</sup>. Bunun sonucu olarak, çıktı katmanından girişe doğru **geri yayılırken gradyan giderek zayıflar** ve özellikle ağın ilk (sığ) katmanlarına neredeyse hiç sinyal ulaşmaz <sup>13</sup>. Sığ katmanlardaki ağırlıklar neredeyse donmuş (hiç güncellenmiyor) hale gelir ve derin SNN’lerin öğrenmesi sekteye uğrar <sup>13</sup>. Bu durum, yapay sinir ağlarında görülen klasik “kaybolan gradyan” problemine benzer, ancak SNN’lerde **diken aktivasyonlarının 0-1 aralığında olması nedeniyle daha da şiddetlidir**. Nitekim, makalenin yazarları **teorik olarak ve deneylerle SNN’lerde gradyan kaybolmasının ne denli ciddi olduğunu göstermektedir**

<sup>14</sup>.

**Makalenin Amacı ve Yaklaşımı:** Bu makalede Guo ve arkadaşları, **SNN eğitiminde gradyan kaybolması sorununu kökten hafifletmeyi** amaçlamaktadır. Bunun için önerdikleri yaklaşımın temel fikri, **kaybolan gradyana “kestirme bir yol” açmaktır**. Yani, çıkış katmanındaki hata bilgisinin (kaybın gradyanının) ara katmanlara **doğrudan aktarılabilceği kestirme geri bağlantılar (shortcut) eklemek** <sup>15</sup>. Bu sayede, sığ katmanlar klasik geri yayılımın getirdiği zayıflamış sinyal yerine doğrudan güçlü bir hata sinyali alır ve ağırlıkları etkin bir şekilde güncellenebilir <sup>16</sup>. Üstelik önerilen “kestirme

**geri yayılım” yöntemi**, yalnızca eğitim esnasında aktiftir; **çıkarım (inference)** sırasında bu ek dallar kaldırılır ve dolayısıyla modele **hiçbir ek yük getirmez** <sup>17</sup> .

Makalede ayrıca, bu yaklaşımı en verimli hale getirmek için bir **“evrimsel eğitim” stratejisi** tanıtılmaktadır. Eğitim sürecinin başında sığ katmanlara ağırlık vermek, sonunda ise asıl çıkışa odaklanmak üzere **dinamik bir denge katsayısı** kullanılır <sup>18</sup> . Bu **kademeli denge değişimi**, ağırlık ilk başta alt katmanlarını yeterince öğrenmeye zorlamasını, sonrasında ise nihai doğruluğu artırmaya odaklanmasını sağlar. Kısacası, makalenin katkısı **SNN eğitime yeni bir bakış** getirerek, **gradyan kaybolması sorununu adresleyen ve pratikte yüksek doğrulukla sonuçlanan bir yöntem** sunmaktır.

## Önceki Çalışmalar (Related Work)

**SNN Öğrenme Paradigmaları:** SNN'ler için literatürde üç temel öğrenme yaklaşımı vardır <sup>19</sup> <sup>20</sup> : 1. **Denetimsiz öğrenme (unsupervised)** – Örneğin, *Spike-Timing-Dependent Plasticity (STDP)* kuralı ile yalnızca yerel zamansal kurallar kullanılarak ağırlık güncellenir <sup>21</sup> . Bu biyolojik olarak tutarlı olsa da, **açık bir hedef (süpervizyon) olmadığı için genellikle küçük ölçekli problemlerle sınırlıdır** ve büyük veri kümelerinde yüksek performans elde etmek zordur <sup>22</sup> . 2. **ANN'den SNN'ye dönüştürme (ANN2SNN)** – Bu yaklaşımda önce klasik bir yapay sinir ağı (ANN) eğitilir, ardından eğitilen ağırlıklar SNN'ye aktarılır ve ReLU gibi aktivasyonlar yerine zamansal spike üreten nöronlar konur <sup>23</sup> . Bu yöntem bazı görsel görevlerde işe yarasa da, **özellikle zamana duyarlı (sekans) verilerde başarısızdır**, çünkü ANN'deki ReLU nöronu sekans bilgisini taşıyamaz <sup>24</sup> . Ayrıca, çok sayıda zaman adımı (ör. binlerce adım) gerekmesi gibi pratik zorlukları da olabilir. 3. **Doğrudan denetimli öğrenme (supervised)** – Bu en yaygın güncel yaklaşımda, SNN doğrudan hataya duyarlı bir yöntemle eğitilir; ancak yukarıda bahsedildiği gibi türev probleminden ötürü **ikame gradyanlar kullanılır** <sup>25</sup> . Denetimli yaklaşım, **az sayıda zaman adımıyla bile büyük veri kümelerinde iyi performans** verebildiği ve **zamansal verileri de işleyebildiği** için giderek daha popüler hale gelmektedir <sup>26</sup> . Bu makaledeki çalışma da **denetimli SNN öğrenimi** kategorisindedir.

**SNN Eğitimindeki Zorlukları Azaltmaya Yönelik Çalışmalar:** Denetimli SNN eğitiminde yaygın olarak kullanılan ikame gradyan (SG) yönteminin kendi getirdiği bazı güçlükler literatürde ele alınmıştır. İlk olarak, **gerçek gradyan ile ikame gradyan arasındaki uyumsuzluk** (gradient mismatch) sorunu vardır <sup>27</sup> . İkame fonksiyonlar, gerçek türevi yalnızca yaklaşık olarak taklit ettiğinden, eğitim sırasında **daha yavaş yakınsama ve daha düşük doğruluk** görülmesi muhtemeldir <sup>27</sup> . Bunu çözmek için önerilen yaklaşımlardan biri **IM-Loss** yöntemidir; bu yöntemde ikame gradyanın şekli eğitim boyunca manuel olarak değiştiriliyor, böylece hem yeterli ağırlık güncellemesi hem de doğruya yakın gradyan sağlanmaya çalışılıyor <sup>28</sup> . Alternatif olarak, **Differentiable Spike** veya **differentiable SG search** gibi yöntemler, en iyi ikame gradyan formunu otomatik olarak arayarak gradient mismatch problemini azaltmayı hedeflemiştir <sup>29</sup> .

İkinci ve bu makale için asıl önemli sorun ise **gradyan kaybolmasıdır**. Tüm ikame fonksiyonlar sınırlı çıktılar ürettiği için türevleri de sınırlıdır ve genellikle giriş uzayının büyük bir bölümünde **gradyan neredeyse sıfırdır** <sup>30</sup> . Bu da SNN'lerde gradyanın katmanlar boyunca hızla küçülmesine ve sığ katmanlarda neredeyse tamamen kaybolmasına yol açar <sup>31</sup> . Bu sorunu hafifletmek üzere literatürde çeşitli çözümler denenmiştir:

- **Derin ağ mimarileri:** *ResNet* benzeri atlamalı bağlantılar, derin ağlardaki gradyan sorunlarını çözmekte etkilidir. Nitekim, SNN'ler için **SEW-ResNet** (toplama öncesi aktivasyon kullanan) ve **MS-ResNet** (önışlemeli aktivasyon kullanan) gibi mimariler önerilmiştir <sup>32</sup> . Amaç, *ResNet*'in sağladığı kısmi çözümü SNN'ye uyarlayarak gradyan akışını iyileştirmektir. Ancak makalenin de işaret ettiği gibi, **SNN'deki spike aktivasyonunun yapısı gereği** sırf bu mimari değişiklikler

gradyan kaybolmasını tamamen önleyememektedir <sup>33</sup> <sup>34</sup> . Zira ResNet'in ANN'de başarılı olma sebebi, ReLU aktivasyonunun pozitif bölgede sınırsız büyüyebilmesi ve geriye anlamlı gradyan iletmesidir; SNN'de ise spiking nöronun çıktısı 0-1 aralığında kalır ve ikame gradyanlar her adımda zayıflar <sup>35</sup> .

- **Normalizasyon teknikleri:** Gradyan kaybolması ya da patlaması sorunlarını hafifletmek için çeşitli **katman normları ve özel normalizasyonlar** kullanılmıştır. Örneğin, **tdBN (eşik bağımlı BatchNorm)** yönteminde her zaman adımında ve kanalda veriler ayrı ayrı normalize edilerek spike dağılımlarının dengelenmesi hedeflenmiştir <sup>36</sup> . Benzer şekilde **BNTT (zamansal BatchNorm)**, **PSP-BN (post-sinaptik potansiyel BN)** ve **TEBN (zamansal etkin BN)** yöntemleri, farklı zaman adımlarındaki spike dağılımlarının tutarsızlığını giderip her bir zaman adımını ayrı normalize ederek gradyan akışını istikrarlı hale getirmeye çalışmışlardır <sup>37</sup> . **MPBN** adlı yöntem ise, her katmanda membran potansiyeli güncellemesinden sonra ek bir batch normalization uygulayarak veri akışını yeniden ölçeklemiştir <sup>38</sup> . Tüm bu yöntemler, gradyan kaybolması ve patlaması problemlerine karşı SNN'lerin eğitimi kararlı kılmayı amaçlar.
- **Düzenleştirme (regularization) yaklaşımları:** Bazı çalışmalar, kaybolan/patlayan gradyan sorununu kayıp fonksiyonuna ek terimler ekleyerek azaltmıştır. Örneğin, **RecDis-SNN** çalışması, nöronların membran potansiyel dağılımını kontrol altında tutmak için bir *potansiyel düzenleme kaybı* tanımlamıştır <sup>39</sup> . **Spiking PointNet** ise "küçük ağ ile eğit, büyük ağ ile kullan" şeklinde özetlenebilecek bir yaklaşım benimsemiş; eğitim sırasında daha küçük bir yardımcı ağ kullanarak eğitimi kolaylaştırıp sonrasında asıl ağı eğitme stratejisi önermiştir <sup>40</sup> . Bu da dolaylı olarak derin ağın eğitim zorluğunu azaltmayı hedefler.

Yukarıdaki yöntemler **kısmen başarılı olsa da**, tümünün ortak bir sınırlaması vardır: **Gradyanı yine katman katman iletme zorundadırlar** <sup>41</sup> . Yani, en nihayetinde hata sinyali ilk katmana ulaşana dek her ara katmanda zayıflamaya devam eder. Bu nedenle, **gradyan kaybolması tamamen ortadan kalkmış değildir** <sup>41</sup> . İşte bu noktada, ele aldığımız makale **alışılmış yaklaşımdan radikal bir sapma** öneriyor: **Gradyanı ara adımlara gerek kalmadan doğrudan sığ katmanlara iletme**. Bu sayede, önceki çalışmaların çözemediği gradyan sönümlenmesi sorununun kökten çözülebildiği iddia ediliyor.

## Önerilen Yöntem (Methodology)

Makalenin önerdiği çözüm, iki ana bileşenden oluşmaktadır: (1) **Kestirme Geri Yayılım Yöntemi** ve (2) **Evrimsel Eğitim Çerçevesi**. İlk bileşen, gradyanın ağ içerisinde iletiliş şeklini değiştirerek kaybolmayı engellerken, ikinci bileşen eğitim sürecindeki odak noktalarını dinamik olarak ayarlayarak nihai doğruluğu maksimize etmektedir.

### Kestirme Geri Yayılım Yöntemi

**Doğrudan Gradyan İletimi:** Kestirme geri yayılım yaklaşımında, klasik derin ağ eğitimine **ek bir kısayol mekanizması** eklenir. Normalde bir derin ağda hata sinyali (loss gradyanı), son katmandan önceki katmana, oradan bir öncekine şeklinde sıralı olarak geriye doğru yayılır. Bu yöntemde ise **ağın ara katmanlarına doğrudan çıkışa bağlanan ek dallar (shortcut branch) eklenir** <sup>15</sup> . Örneğin, bir CNN tabanlı SNN'nin her bir blok katmanından sonra bir kestirme çıkışı alınarak doğrudan nihai çıktı katmanına bağlanır (Şekil 1'de görsel bir örnek verilmiştir). Bu ara dalların her biri, **küresel ortalama havuzlama (global average pooling)** ve **tam bağlantılı (fully-connected) bir katman** ile kendi çıkışını (tahminini) üretir <sup>42</sup> . Sonuçta ağın **nihai çıktısı**, ana dal (orijinal son katman) çıktısıyla bu ara dalların çıktılarının bir bileşimi haline gelir <sup>43</sup> . Matematiksel ifade ile, klasik ağın orijinal çıktısı  $O_{\text{main}}$  iken, kestirmeli ağın eğitime özel çıktısı:

$$O_{\text{combined}} = f_{\text{main}}(x) + \sum_{i=1}^{N-1} f_{\text{side}_i}(x)$$

gibi, ara katmanların fonksiyonları  $f_{\text{side}_i}$  ile zenginleştirilir (burada  $N$  toplam blok sayısı). Her bir ara dalga karşılık gelen bloktan aldığı özelliği sınıflandırıp bir **yardımcı tahmin** üretir.

**Gradyan Kaybolmasının Giderilmesi:** Bu yapı sayesinde, **hata sinyali doğrudan bu ara dallar üzerinden sığ katmanlara iletilebilir**. Örneğin, en ilk katmanın ağırlıkları için klasik durumda gradyan hesaplanırken zincir kuralı ile ardışık türevlerin çarpımı yer alır ve her katmanda bu değer küçülür. Oysa kestirme dallar eklendiğinde, **ilk katmanın ağırlıklarına, ana kayıp (loss) fonksiyonundan doğrudan bir türev bileşeni eklenir** <sup>44</sup>. Yazarlar, birinci katman ağırlığı  $W_1$  için gradyanları karşılaştırmalı olarak formüle ediyor:

- **Vanilla (klasik) durumda:**  $\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial O_{\text{main}}} \cdot \frac{\partial O_{\text{main}}}{\partial W_1}$  (burada  $\frac{\partial L}{\partial O_{\text{main}}}$  ifadesi, tüm ara katmanların türev zincirini içerir). İkame türevler nedeniyle bu değer **çok küçük** olur, dolayısıyla  $W_1$  neredeyse hiç güncellenmez <sup>44</sup>.
- **Kestirmeli yöntemde:**  $\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial O_{\text{main}}} \cdot \frac{\partial O_{\text{main}}}{\partial W_1} + \frac{\partial L}{\partial O_{\text{side}_1}} \cdot \frac{\partial O_{\text{side}_1}}{\partial W_1}$ . Burada ikinci terim,  $W_1$ 'in etkilediği ilk ara dalın doğrudan katkısını temsil eder <sup>45</sup>. Bu sayede  $W_1$ 'e ulaşan toplam gradyan değeri  $\frac{\partial L}{\partial W_1}$  **kayda değer ölçüde büyüktür** ve ilk katman dahil tüm sığ katmanlar yeterli derece güncellenebilir hale gelir <sup>46</sup>.

Yazarlar, önerdikleri yöntemin etkisini göstermek için deneysel bir analiz de sunmuşlardır: Kestirme dallar kullanılarak eğitilen bir SNN'de, **ilk katmandaki gradyan dağılımının oldukça düzgün ve yayvan olduğunu** (yani 0 etrafında yığılmadığını) göstermişlerdir <sup>47</sup>. Bu, gradyan kaybolması sorununun sığ katmanlar için **tamamen çözüldüğüne** işaret etmektedir. Önemle belirtmek gerekir ki, **bu kestirme dallar sadece eğitim sırasında aktiftir**. Eğitim tamamlandıktan sonra ara dallar kaldırılır ve ağ, normal bir SNN yapısı olarak çıkarım yapar <sup>47</sup>. Yani, **inference aşamasında ek bir hesaplama maliyeti veya parametre yükü getirmez** <sup>47</sup>. Bu özellik, yöntemin pratik açıdan çok önemli bir avantajıdır; zira enerji verimli SNN'lerin temel amacı, çıkarımda tasarruf sağlamak olduğundan, eğitimdeki ek mekanizmaların çıkarım zamanında engel oluşturmaması gerekir.

## Evrimsel Eğitim Çerçevesi

Kestirme geri yayılım yöntemi tek başına gradyan akışını iyileştirerek eğitim performansını artırsa da, beraberinde yeni bir **optimizasyon ikilemi** getirir. Her bir ara daldan gelen çıkışlar için ağ, adeta **çoklu kayıp fonksiyonlarını ortaklaşa optimize eden** bir hale gelmiştir <sup>48</sup>. Bu durumda, hangi kayba (ana dalın kaybı vs. yan dalın kayıpları) ne kadar ağırlık verileceği kritik bir sorundur:

- **Ana dalın (son katmanın) kaybına yüksek ağırlık verirsek**, ağ büyük oranda orijinal çıktı doğruluğuna odaklanır, ancak bu durumda **sığ katmanların güncellenmesi için kestirme dalların pek bir etkisi kalmaz**. Yani gradyan kaybolmasını önleme avantajı azalır <sup>18</sup>.
- **Yan dalların (ara katman çıkışlarının) kayıplarına yüksek ağırlık verirsek**, bu sefer de ağ sığ katmanları öğrenmeye fazlasıyla odaklanır ancak **nihai doğruluk düşebilir**, çünkü sonuçta test zamanında yan dallar yok ve sadece ana çıkış önemlidir <sup>18</sup>.

Yazarlar bu denge problemine karşı **"evrimsel (aşamalı) eğitim"** adını verdikleri bir çözüm getiriyorlar. Bu strateji, **eğitim süresince odak dağılımını zamanla değiştirmeyi** içeriyor <sup>49</sup>. Temel fikir: **Eğitimin**

**başlarında yan dallara (sığ katmanların çıktıları) daha fazla önem ver**, sonlarına doğru giderek **ana dalın çıktısına daha fazla önem ver** <sup>49</sup> . Bu sayede başlangıçta ağıın sığ katmanları iyice öğreniyor, sonlarda ise ana ağıın performansı iyileştiriliyor.

Bunu uygulamak için, modelde  $\lambda$  adında bir “denge katsayısı” tanımlanıyor.  $\lambda$  değeri **her eğitim iterasyonunda (veya epoch’unda) sistematik olarak güncellenerek** yan ve ana kayıp arasında bir ağırlıklandırma yapıyor <sup>50</sup> . Örneğin, eğitimin en başında  $\lambda$  küçük bir değer (yazarlar 0.25 olarak belirlemiş <sup>51</sup> ), böylece **ana dal kaybı =  $\lambda L_{\text{main}}$ , tüm yan dal kayıpları =  $(1-\lambda) L_{\text{side-total}}$**  gibi düşünülebilir (burada  $L_{\text{side-total}}$  tüm yan dalların kayıplarının toplamı veya ortalamasıdır).  $\lambda$  küçükken  $(1-\lambda)$  büyük olur, yani ağırlığın çoğu yan dallara gider – ağı bu aşamada sığ katman hatalarını minimize etmeye çalışır. Eğitim ilerledikçe  $\lambda$  giderek artırılır (önerilen formüle göre doğrusal veya üstel bir artış olabilir) <sup>52</sup> . Son epoch geldiğinde  $\lambda \approx 1$  olur; yani artık **yalnızca ana dalın hatası** önemli hale gelir, yan dallar neredeyse tamamen göz ardı edilir. Bu geçiş sırasında sığ katmanlar yeterince öğrenmiş olduğundan, eğitim sonuna doğru ağı **nihai doğruluğu cilalamaya** odaklanır.

Bu yaklaşım evrimsel olarak adlandırılıyor çünkü **eğitim dinamik bir süreç gibi ele alınıyor**: önce ağıın “genç” döneminde farklı bir hedef, olgunlaştıkça farklı bir hedef güdülüyor. Makalede bu sürecin akışı **Algoritma 1** ile belirtilmiş; özünde her iterasyonda veri alınması, ileri yayılımı tüm dalların çıktı üretilmesi,  $\lambda$ ’ın güncellenmesi, bileşik çıktının hesaplanması, kaybın ve gradyanların hesaplanıp ağırlıkların güncellenmesi adımları tekrarlanıyor <sup>53</sup> <sup>54</sup> . Çıkarım aşamasında ise  $\lambda$  anlamsız, çünkü yan dallar zaten yok – dolayısıyla sadece ana ağıın çıktısı hesaplanıp sonuca bakılıyor <sup>55</sup> <sup>56</sup> .

**Özetle**, evrimsel eğitim çerçevesi, **kestirme dalların getirdiği faydayı nihai doğruluktan ödün vermeden kullanabilmek için** bir çözüm sunar. Başlangıçta **öğrenmesi zor olan kısımları (sığ katmanları) güçlendirir**, sonunda ağı **asıl hedefe yönlendirir**. Bu ikili stratejiyle, yöntem saf kestirme geri yayılıma kıyasla **birkaç ekstra puanlık doğruluk kazanımı** sağlayabilmektedir (Aşağıdaki deney sonuçlarında bu görülecektir). Sonuç olarak, makalenin önerdiği yöntem bir yandan **gradyan kaybolmasını kökten çözen kestirme geri yayılım**, diğer yandan **eğitimi doğru odakla yöneten evrimsel planlama** birleşiminden oluşmaktadır.

## Deneyler (Experiments)

Yazarlar, önerdikleri yöntemin etkinliğini göstermek için **kapsamlı deneyler** gerçekleştirmişlerdir. Deneyler, hem **statik görüntü veri setlerinde** (CIFAR-10, CIFAR-100, ImageNet) hem de **dinamik “olay temelli” veri setinde** (CIFAR10-DVS) yürütülmüştür <sup>57</sup> . Farklı ölçeklerdeki bu veri kümeleri sayesinde yöntemin küçük ve orta ölçekli problemlerden büyük ölçekli ve zamansal problemlere kadar performansı değerlendirilmiştir. Model mimarileri olarak yaygın kullanılan **Residual Network (ResNet)** türevleri seçilmiştir (örn. ResNet18, ResNet34 gibi). Eğitim sırasında, adil bir karşılaştırma için tüm yöntemler benzer veri ön işleme ve **veri artırma teknikleri** ile değerlendirilmiştir – örneğin CIFAR için standart normalizasyon, rastgele kırma, yatay çevirme ve AutoAugment kullanılmıştır <sup>58</sup> <sup>59</sup> . CIFAR10-DVS verisi ise literatürdeki protokole uygun şekilde eğitim/teste bölünmüş ve boyutları ayarlanmıştır <sup>60</sup> .

**1. Ablasyon Çalışması (Kestirme & Evrimsel Eğitimin Etkisi)**: Öncelikle önerilen yöntemlerin tek tek ve bir arada katkısını görmek amacıyla **CIFAR-100 üzerinde ablatif deneyler** yapılmıştır <sup>61</sup> . Bu

deneylerde ResNet18 (daha sığ) ve ResNet34 (daha derin) modelleri, **farklı zaman adımı (time-step) sayılarıyla** eğitilerek karşılaştırılmıştır. Sonuçlar **Tablo 1** ile özetlenmiştir. Başlıca bulgular:

- **ResNet18** için klasik eğitimle elde edilen doğruluk, 2 zaman adımı için %71.42, 4 zaman adımı için %72.22 olarak raporlanmıştır <sup>62</sup>. Bu değerler, literatürdeki benzer yöntemlerle uyumludur ve bir nevi temel seviye performansı temsil etmektedir. Makalenin **kestirme geri yayılım** yöntemini uyguladıklarında ise aynı ağırlık doğruluğu 2 adımda %73.68'e, 4 adımda %74.78'e yükselmiştir <sup>63</sup>. Yani yaklaşık **2.5 puanlık** anlamlı bir iyileşme görülmektedir. Buna ek olarak, **evrimsel eğitim** stratejisi de dahil edildiğinde doğruluk biraz daha artarak 2 adımda %74.02'ye, 4 adımda %74.83'e çıkmıştır <sup>64</sup>. Evrimsel stratejinin tek başına katkısı ResNet18 için sınırlı (~+0.2-0.3 puan) olsa da toplamda baz eğitime göre **%3'e yakın bir iyileşme** elde edilmiştir.
- **ResNet34** için sonuçlar daha da çarpıcıdır. Klasik (vanilla) eğitimle ResNet34, 2 adımda %69.82, 4 adımda %69.98 doğruluk elde etmiştir <sup>65</sup>. İlginç biçimde, bu değerler daha küçük bir ağırlık olan ResNet18'in gerisindedir. Yazarlar bu durumu, **derin SNN'lerde gradyan kaybolmasının yarattığı zorlukla** açıklamaktadır: derinlik arttığında, sığ katmanlar neredeyse hiç öğrenememekte ve derin modelin potansiyeli kullanılamamaktadır <sup>66</sup>. Ancak **kestirme geri yayılım yöntemi uygulandığında**, ResNet34'ün performansı dramatik biçimde yükselmiştir – 2 adımda %74.06, 4 adımda %75.67 doğruluk elde edilmiştir <sup>67</sup>. Bu, derin modele **~5 puanlık** muazzam bir kazanç sağlamış ve **ResNet18'i de geride bırakmasını** mümkün kılmıştır <sup>67</sup>. Yani yöntem, özellikle **derin ağlarda kaybolan gradyanı telafi ederek derinliğin getireceği performans artışını ortaya çıkarabilmiştir**. Ayrıca ResNet34'e evrimsel eğitim de uygulandığında doğruluk bir miktar daha artarak 2 adımda %74.17, 4 adımda %75.81 seviyesine gelmiştir <sup>68</sup>. Bu sonuçlar, önerilen her iki bileşenin de (özellikle kestirme gradyan iletiminin) **etkili olduğunu net bir şekilde göstermektedir** <sup>69</sup>.

Özetle, ablatif deneyler **yöntemin başarısını doğrulamaktadır**: Kestirme gradyan iletimi tek başına ciddi kazanımlar sağlarken, evrimsel ağırlıklandırma stratejisi de ekstra iyileşme getirmiştir. Özellikle **derin modellerin normalde başarısız olduğu bir senaryoda (ResNet34, kaybolan gradyan nedeniyle ResNet18'den kötü) yöntemin tabloyu tersine çevirdiği** görülmüştür <sup>70</sup>. Bu, makalenin ana iddiasını – gradyan kaybolması sorununun giderilmesinin derin SNN'lerin potansiyelini açığa çıkaracağı – doğrular niteliktedir.

**2. Güncel Yöntemlerle Karşılaştırma:** Yazarlar yöntemlerini dönemin **en iyi yöntemleriyle (SoTA - state-of-the-art) karşılaştırmak** için kapsamlı deneyler yapmışlardır. Bu kapsamda hem **CIFAR-10 ve CIFAR-100** için, hem de **ImageNet** ve **CIFAR10-DVS** için ayrı karşılaştırmalar sunulmuştur. Bütün karşılaştırmalarda adil şartlar sağlamak adına sonuçlar **3 bağımsız çalıştırmanın ortalaması** olarak rapor edilmiştir <sup>71</sup>.

- **CIFAR-10 ve CIFAR-100:** Tablo 2'de bu veri kümeleri üzerindeki sonuçlar listelenmiştir <sup>72</sup> <sup>73</sup>. Karşılaştırılan yöntemler arasında **ANN2SNN türü** (örn. SpikeNorm, RMP), **hibrit eğitim** (Hybrid-Train), **tandem öğrenme** (TL, PTL, LTL gibi) ve doğrudan SNN eğitim yöntemleri (TSSL-BP, PLIF, DSR, KDSNN, Diet-SNN, Dspike, STBP-tdBN, TET, RecDis-SNN, Real Spike vb.) bulunmaktadır <sup>74</sup> <sup>75</sup>. Bu geniş yelpazede, her bir yöntem farklı mimari ve zaman adımı kombinasyonlarıyla en iyi sonuçları hedeflemiştir. Örneğin, CIFAR-10'da **Real Spike yöntemi** ResNet19 ile 4 zaman adımında %95.51 doğruluk alarak önceki en iyi değeri sağlamıştır; yine **DSR yöntemi** ResNet18 ile 20 adımda %95.40 gibi yüksek bir sonuca ulaşmıştır <sup>76</sup>. **TET yöntemi** 4 adımda %94.44, **RecDis-SNN** 4 adımda %95.53 gibi oldukça rekabetçi sonuçlar elde etmiştir. Bu sonuçlar, klasik ANN performanslarına yakın değerlere SNN'lerle ulaşılabilirliğini gösterir.

Önerilen **Kestirme Geri Yayılım** yöntemi ise ResNet18 omurgasıyla 1 adımda %93.89, 2 adımda %93.92 gibi dikkat çekici doğruluklar elde etmiştir <sup>77</sup> <sup>78</sup> . Sadece **tek zaman adımı** ile dahi yaklaşık %93.9 doğruluk almak, SNN'lerde **aşırı düşük gecikme (latency) ile yüksek doğruluk** sağlamak açısından önemlidir. Zaman adımı 4'e çıkarıldığında SBP yöntemiyle %94'ü aşan sonuçlar görülmüştür <sup>79</sup> . Yazarlar, kendi ağlarının ResNet18 (11.2M parametre) olduğunu, oysa en iyi sonuçlardan bazılarının ResNet19/20 (11.5M/11.1M parametre) ile alındığını belirterek, benzer ölçekteki ağlarla kıyaslandığında yöntemlerinin oldukça rekabetçi olduğunu vurgulamışlardır <sup>76</sup> . Dahası, **evrimsel eğitim çerçevesi eklendiğinde** performans daha da yükselmiş; ResNet18 ile 4 adımda yaklaşık **%94.46** doğruluk elde edilmiştir (Evrimsel eğitilmiş SNN, 4 adım) <sup>80</sup> <sup>81</sup> . Bu değer, örneğin TET (ResNet19, 4 adım, %94.44) ve Dspike (ResNet20, 4 adım, %93.66) gibi yöntemlerle **benzer veya daha iyi seviyededir**. Yani, **daha az zaman adımı kullanarak** bu yüksek doğruluklara ulaşılabilmektedir. CIFAR-100 veri kümesinde ise yöntemlerinin en iyi sonucu, ResNet18 ile 4 adımda **%74.83** olarak raporlanmıştır <sup>82</sup> . Bu değer, o dönemdeki en iyi yöntem olan TET'in ResNet19 ile elde ettiği %74.47'lik sonucu yaklaşık **%0.4-0.5 oranında geçmektedir** <sup>83</sup> . Yine RecDis-SNN'in %74.10'luk sonucunun da belirgin şekilde üzerindedir. Önemle belirtmek gerekirse, yazarların yöntemi bunu **daha küçük bir ağ ve benzer veya daha az zaman adımı** ile başarmıştır. Bu sonuçlar, **önerilen yaklaşımın hem verimli hem de etkili** olduğunu açıkça göstermektedir <sup>83</sup> .

- **ImageNet:** Bu, SNN'ler için oldukça zorlu ve büyük ölçekli bir veri kümesidir. Makalede, ImageNet üzerinde de karşılaştırmalar **Tablo 3** ile verilmiştir <sup>84</sup> <sup>85</sup> . Karşılaştırılan yöntemlerden bazıları: RecDis-SNN, GLIF, DSR, MPBN, MS-ResNet, Real Spike, SEW-ResNet gibi güncel çalışmalardır <sup>86</sup> . Bu yöntemlerin çoğu, ResNet34 veya ResNet18 tabanlı modellerle 4 ila 6 zaman adımı kullanarak yaklaşık %63-67 arası doğruluklara ulaşmışlardır <sup>87</sup> <sup>88</sup> . Örneğin **RecDis-SNN** (ResNet34, 6 adım) %67.33, **GLIF** (ResNet34, 4 adım) %67.52, **Real Spike** (ResNet34, 4 adım) %67.69 gibi sonuçlar almıştır. Dikkat çekici bir nokta, Real Spike ve SEW-ResNet gibi bazı yöntemlerin, **ara katmanlarda tam sayı değerli çıktılar üreterek** performansı artırdığı ancak bunun donanım üzerinde ekstra enerji maliyeti yarattığı belirtilmiştir <sup>89</sup> . Yani bu yöntemler, klasik ResNet mimarisinden saparak enerji verimliliğinden bir miktar ödün vermektedir. Buna karşılık, **önerilen yöntem standart ResNet mimarisi** (ResNet18 ve ResNet34) ile çalışır ve yine de yüksek performans elde eder <sup>90</sup> . Sonuçlara bakıldığında, **SBP yöntemi** ResNet18 ile 4 adımda %64.47, ResNet34 ile 4 adımda %67.90 doğruluk sağlamıştır <sup>91</sup> . **Evrimsel eğitim eklenince** bu değerler ResNet18 için %65.12, ResNet34 için **%68.14** seviyesine çıkmıştır <sup>92</sup> . Özellikle ResNet34 + Evo kombinasyonu, %68.14 ile listelenen tüm yöntemler arasında en yüksek değere ulaşmaktadır. Örneğin, bu sonuç Real Spike (ResNet34, 4 adım, %67.69) değerinden **+0.45 puan**, SEW-ResNet'ten (ResNet34, 4 adım, %67.04) **+1.1 puan** daha yüksektir <sup>93</sup> . Ayrıca RecDis-SNN'in 6 adımda %67.33 değeri de geride bırakılmıştır. Yazarlar bu başarıyı vurgulayarak, yöntemlerinin **büyük ölçekli bir veri setinde dahi etkili olduğunu** ve standart mimarilerle bile mevcut en iyi yöntemleri aşabildiğini belirtmişlerdir <sup>93</sup> .

- **CIFAR10-DVS (Nöromorfik veri):** Bu veri kümesi, DVS (Dynamic Vision Sensor) denilen olay temelli kamera verisinden türetilmiştir ve SNN'lerin doğal çalıştığı bir ortam sunar. Tablo 4'te, 10 zaman adımı kullanılarak eğitilen farklı yöntemlerin sonuçları verilmiştir <sup>94</sup> . Önceki en iyi yöntemler arasında **Dspike** (ResNet18, 10 adım, %75.40), **Real Spike** (ResNet19, 10 adım, %72.85) ve **RecDis-SNN** (ResNet19, 10 adım, %72.42) bulunmaktadır <sup>95</sup> . Önerilen yöntem ise ResNet18 ile **10 adımda %82.00** doğruluk elde etmiştir; evrimsel eğitim eklendiğinde bu değer **%83.30** gibi son derece yüksek bir seviyeye ulaşmıştır <sup>96</sup> <sup>97</sup> . Bu, önceki en iyi sonucu yaklaşık **%8-8.5 puan** geliştirmek anlamına gelir ki nöromorfik verilerde bu büyüklükte bir sıçrama son derece kayda değerdir. Yazarlar özellikle bu sonucu vurgulayarak, yöntemlerinin **nöromorfik gerçek zamanlı uygulamalarda performansı çarpıcı biçimde artırdığını** belirtmişlerdir <sup>98</sup> . Enerji verimliliğiyle öne çıkan SNN'lerin, bu yöntem sayesinde **çok daha yüksek doğruluklara ulaşarak pratik uygulanabilirliğinin arttığı** ifade edilebilir.

Genel olarak deneysel sonuçlar, **önerilen yaklaşımın tutarlı bir şekilde iyileştirme sağladığını** gösteriyor. Farklı ağ derinliklerinde, zaman adımı sayılarında ve veri tiplerinde SBP + evrimsel eğitim yöntemi, ya önceki yöntemlerle başa baş performans sergilemiş ya da onları geride bırakmıştır. Özellikle **derin SNN'lerin eğitimi** ve **çok düşük adım sayısı ile yüksek başarıya ulaşılması** konularında önemli bir ilerleme kaydedilmiştir. Bu da makalenin temel motivasyonunun – derin SNN'leri daha erişilebilir kılmak – başarıldığını göstermektedir.

## Analizler ve Tartışma (Analysis & Discussion)

Makale boyunca sunulan **teorik ve deneysel analizler**, önerilen yöntemin neden ve nasıl işe yaradığına dair çeşitli içgörüler sunmaktadır:

- **Gradyan Dağılım Analizi:** Yazarlar, kestirme geri yayılımın etkisini somutlaştırmak için **gradyanların büyüklük dağılımlarını** incelemişlerdir. Özellikle, CIFAR-10 üzerinde eğitilen Spiking ResNet-20 modelinin ilk katmanındaki gradyanları görselleştirmişlerdir <sup>47</sup>. Klasik ikame gradyan yöntemi kullanıldığında, bu dağılımın **0 civarında çok sivri** (yani çoğu gradyan değeri 0'a yakın) olduğu; dolayısıyla ilk katman ağırlıklarının neredeyse sabit kaldığı gözlemlenmiştir. Buna karşılık, önerilen kestirme yöntemi uygulandığında dağılım **daha düz ve geniş bir yayılım** göstermiştir <sup>47</sup>. Bu, farklı örnekler ve gradyanlar boyunca ilk katmana anlamlı büyüklükte hata sinyalleri ulaştığını, yani gradyan kaybolmasının etkin biçimde önlendiğini gösterir. Bu deney, matematiksel çıkarımlarını görsel olarak da doğrulayarak yöntemin etkisine dair güvenilirlik sağlamaktadır.
- **Derinlik ve Model Kapasitesi:** Ablasyon deneylerinde görüldüğü gibi, **derin SNN'ler normalde beklenen performansı gösteremeyebilir**. ResNet34'ün ResNet18'den kötü sonuç vermesi, klasik eğitimde gradyan kaybının derin modelleri adeta “kullanılamaz” hale getirebildiğini gösterdi <sup>70</sup>. SBP yöntemi, bu engeli kaldırarak derin modellerin daha iyi genelleşmesini sağladı. Bu durum, **SNN alanında daha derin mimarilere yönelimin önünü açabilir**. Önceden, gradyan sorunları nedeniyle daha derin ağlar anlamsız iken, şimdi SBP ile 50-katmanlı, 100-katmanlı SNN'leri eğitmek mümkün hale gelebilir. Ayrıca, sonuçlar göstermektedir ki **daha derin ağlar SBP'den daha fazla fayda sağlıyor** (ResNet34, SBP ile +5% artış alırken ResNet18 +2.5% almıştı) <sup>67</sup>. Bu da yöntemin özellikle kompleks mimariler için değerli olduğunu ortaya koyar.
- **Zaman Adımı Etkisi:** SNN'lerde doğruluk genellikle zaman adımı (farklı zaman dilimlerinde girişin işlenmesi) arttıkça yükselir, ancak bu da gecikme ve hesaplama maliyetini artırır. SBP yöntemi, **daha az zaman adımıyla** yüksek doğruluk elde etmeyi kolaylaştırmıştır. Örneğin CIFAR-100'de sadece 2 adım ile %74.8 doğruluk elde edilmiştir ki, bu değer daha önce ancak daha fazla adımla ulaşılabilen bir seviyedir <sup>99</sup> <sup>100</sup>. Hatta yazarlar, 2 zaman adımıyla ResNet19 kullanarak %77.79 doğruluk elde ettiklerini ve bunun daha fazla adım kullanan SNN modellerinden bile ~%3.3 daha iyi olduğunu belirtiyorlar <sup>99</sup>. Bu, SBP'nin **zaman boyutunda da verimlilik** sağladığını gösterir. Az sayıda adım, demek ki her adımın taşıdığı bilgi daha verimli kullanılabilir; çünkü gradyanlar doğru yerlere ulaşarak ağı etkin şekilde eğitmiş oluyor. Pratik anlamda bu, **daha düşük gecikme ile çalışan SNN tabanlı sistemler** kurulabileceği anlamına gelir (örneğin gerçek zamanlı uygulamalarda hızlı tepki için SNN adım sayısını düşük tutmak çok kritiktir).
- **Evrimsel Stratejinin Katkısı:** Evrimsel eğitim stratejisinin devreye sokulmasıyla genelde %0.2–0.5 aralığında ek doğruluk kazanımları görülüyor. Bu görece küçük artışlar, belki SBP kadar dramatik değil, ancak yazarların vurguladığı önemli bir noktayı destekliyor: **yan dallar ile ana dal arasında uygun dengeyi kurmak nihai performans için kritik**. Özellikle yan dalların çok



baskın olduğu bir eğitim, eğer evrimsel olarak ana dala tekrar yönelmeseydi, muhtemelen en sonunda küçük bir genel doğruluk kaybına yol açabilirdi. Evrimsel strateji bunu önleyerek, “hem havuç hem sopa” misali ağıın önce bir kısmını sonra diğer kısmını optimize etmesini sağlıyor. Elde edilen sonuçlar da gösteriyor ki, **SBP yöntemine bu stratejiyi eklemek daima daha iyi ya da en kötü eşit performans veriyor**, hiçbir durumda zarar vermiyor. Bu da bu yaklaşımın uygulanabilir ve güvenli bir ek adım olduğunu gösterir.

- **Enerji ve Maliyet Analizi:** Makalede doğrudan enerji ölçümü veya hesaplama maliyeti analizi bulunmuyor, ancak dolaylı olarak bazı noktalar çıkarabiliriz. Birincisi, **SBP yöntemi inference sırasında ek maliyet getirmiyor** <sup>47</sup>. Bu, enerji verimliliği hedefleyen SNN'ler için çok önemli. İkincisi, evrimsel eğitim aşamasında yan dalların ağırlıklandırılması, eğitim süresini biraz artırsa da (her iterasyonda belki ufak ekstra hesaplamalar), eğitim tamamlandığında ortaya çıkan model için herhangi bir ek parametre yok. Bu nedenle, yöntemin getirdiği **hesaplama maliyeti sadece eğitime has** ve bu da genellikle kabul edilebilir bir durum. Ayrıca zaman adımı sayısını düşürerek aynı doğruluğu yakalamak, inference'da yapılacak toplam hesaplamayı da düşürecektir (SNN'lerde her zaman adımı, ağıın tekrar çalıştırılması anlamına geliyor). Yani SBP + evrimsel eğitim, **zaman-adım doğruluk eğrisini daha avantajlı bir noktaya çekerek** pratikte enerji tasarrufuna da katkıda bulunabilir.

Özetlemek gerekirse, makalenin analiz kısmı, **yöntemin hem teorik gerekçelerini hem de deneysel gözlemlerini** ortaya koyarak şu mesajı veriyor: Önerilen yaklaşımla **SNN'lerin kronik bir sorunu çözülmüş** ve bunun sonucunda **daha derin, daha hızlı ve daha doğru SNN'ler** eğitmek mümkün hale gelmiştir. Bu gelişme, SNN araştırmalarında yeni bir sayfa açma potansiyeline sahiptir.

## Sonuçlar (Conclusion)

Makalenin sonucunda yazarlar, çalışmalarını özetleyerek şu önemli noktaların altını çizmişlerdir: Öncelikle, **SNN'lerde gradyan kaybolması probleminin varlığını ve ciddiyetini** hem teorik analizlerle hem de deneysel verilerle ortaya koymuşlardır <sup>101</sup>. Bu sorunu gidermek için, **kestirme geri yayılım** adını verdikleri basit fakat etkili bir yöntem önermişlerdir. Bu yöntem ile **hata gradyanı doğrudan sığ katmanlara iletilerek** gradyan kaybolması önemli ölçüde hafifletilmiştir <sup>101</sup>. Önemlisi, bu çözüm **ağıın çalışma zamanındaki yapısını veya verimliliğini hiç bozmamaktadır** – eklenen dallar tamamen eğitim sürecine özgüdür ve çıkarımda kaldırılır <sup>102</sup>.

Ayrıca, yazarlar **evrimsel eğitim çerçevesi** ile yöntemlerini bir adım ileri taşımışlardır. Dinamik bir denge katsayısı yardımıyla, eğitim boyunca odak noktasını yavaşça ana ağa kaydırarak **hem sığ katmanların iyi öğrenmesini hem de final doğruluğun yüksek olmasını** aynı anda başarmışlardır <sup>103</sup>. Bu çerçevenin de katkısıyla yöntem, çeşitli deneylerde **ağıın doğruluğunu daha da iyileştirmiştir**.

Son olarak, kapsamlı deneylerle **yöntemin etkililiği kanıtlanmıştır**: CIFAR-10, CIFAR-100 gibi statik ve CIFAR10-DVS gibi spiking veri setlerinde, üstelik yaygın mimariler kullanılarak, önerilen yöntemle eğitilen SNN'lerin **tutarlı bir şekilde daha iyi performans** sergilediği gösterilmiştir <sup>101</sup>. Birçok senaryoda mevcut en iyi sonuçları ya yakalamış ya geçmiştir. Özellikle vurgulanan bir sonuç, **2 zaman adımı ile CIFAR-100'de %77.79 doğruluk** elde edilmesidir ki, bu benzer ağların o güne kadarki en iyi sonucundan ~%3.3 daha yüksektir <sup>99</sup>. Bu tür kazanımlar, önerilen yaklaşımın pratik değerini ortaya koymaktadır.

Genel itibarla, **“Take A Shortcut Back” çalışması SNN'lerin eğitimine dair önemli bir ilerleme sunmaktadır**. Gradyan kaybolması gibi temel bir probleme yeni bir bakış açısıyla çözüm getirilmiş ve bu çözümün işe yaradığı kapsamlı bir şekilde gösterilmiştir. Bu sonuçlar, SNN araştırma topluluğu için derin

ağlar eğitme, daha yüksek doğruluk elde etme ve gerçek dünyada SNN uygulamalarını genişletme açısından oldukça değerlidir.

## Makalenin Önemi ve Ortaya Koyduğu Yenilikler

Bu makalenin alanına yaptığı katkıları ve yenilikleri birkaç noktada özetleyebiliriz:

- **Kritik Bir Sorunun Belirlenmesi:** Öncelikle, yazarlar SNN'ler için *"kaybolan gradyan" probleminin* altını çizerek bu sorunu öncelikli bir araştırma konusu haline getirmişlerdir. Bu problem aslında literatürde dolaylı olarak bilinse de, makale bunun **SNN'lerde eğitim derinliğini ciddi şekilde sınırladığını** açıkça ortaya koymuştur <sup>13</sup> <sup>70</sup>. Bu tespit, sonraki çalışmaların da dikkatini çekebilecek önemli bir teorik katmandır. Yani çalışmanın önemi, yalnızca bir yöntem önermek değil, aynı zamanda **SNN'lerin performans tavanını belirleyen bir engeli netleştirmektir**.
- **Yenilikçi Yöntem (Kestirme Geri Yayılım):** Makalenin birincil yeniliği, **gradyan iletimine dair köklü bir değişiklik** öneren kestirme geri yayılım yöntemidir. Klasik derin öğrenmede arka arkaya katmanlardan geçen hata akışı konsepti, burada SNN'lere özgü bir bakışla revize edilmiştir. **Ara katmanlardan doğrudan çıktı bağlantıları eklemek**, literatürde *"deep supervision"* veya *"auxiliary classifier"* olarak başka bağlamlarda görülebilir; fakat SNN'ler için bunu **gradyan problemini çözme amacıyla ve inference maliyeti olmadan** uygulamak özgün bir fikirdir. Bu yöntem **basit ama etkili** oluşuyla da dikkat çekicidir <sup>104</sup>. Gerçekten de, karmaşık ek mekanizmalar yerine birkaç yan dal ekleyerek (ki bunlar halihazırda hesaplanan aktivasyonlardan besleniyor) bu denli iyileşme elde etmek, yöntemin pratikte de kolaylıkla uygulanabileceğini gösterir. Üstelik bu yenilik, **enerji verimliliği hedefine ters düşmeyecek** şekilde tasarlanmıştır – çıkarım sırasında devre dışı kaldığından, SNN'nin düşük güç avantajını korur <sup>104</sup>.
- **Eğitim Stratejisinde Yenilik (Evrimsel Çerçeve):** Yöntemin ikinci yenilikçi yönü, **eğitim dinamiklerinin ayarlanmasına** dair getirdiği yeni bakış açısıdır. Derin öğrenmede eğitim sürecinde loss ağırlıklarının zamanla değiştirilmesi nadiren başvurulan bir tekniktir (ör. *curriculum learning* konsepti kısmen benzer sayılabilir). Burada, SNN'lerin özel durumu için **"önce ağı böyle eğit, sonra şöyle eğit"** şeklinde bir strateji önerilmiştir. Bu **evrimsel eğitim** yaklaşımı, *joint optimization* problemini çözmek için kullanılan yaratıcı bir fikirdir. Bunun yeniliği, **ağ içindeki rekabet halindeki hedefleri (yan çıkışlar vs ana çıkış) zaman-boyutunda dengelemek** şeklinde özetlenebilir. Sonuçlar, bu stratejinin gerçekten de fayda sağladığını gösterdiğinden, gelecekte benzer *aşamalı eğitim* fikirleri başka alanlarda da ilham kaynağı olabilir. Bu yönüyle makale, sadece SNN eğitime değil, genel eğitim stratejilerine de bir katkı sunmaktadır.
- **Performans ve Sonuçlar Açısından Önem:** Makalenin önemi, yalnızca bir problem tanımlayıp çözüm önermesinde değil, **bu çözümün somut kazanımlarla gösterilmesinde** de yatıyor. Elde edilen deney sonuçları, **SNN alanında bazı rekorları kırmış veya yaklaştırmıştır**. Örneğin, CIFAR10-DVS sonuçlarıyla açık ara fark atılması, SNN'lerin potansiyelini gerçek dünya uygulamalarında artırmak demektir <sup>98</sup>. CIFAR-100'de daha az zaman adımıyla SOTA seviyesini geçmek, SNN'lerin verimlilik avantajını korurken performanstan ödün vermeye gerek olmadığını ispatlar <sup>99</sup>. Bu tip sonuçlar, **SNN araştırma topluluğuna moral verici** niteliktedir ve yöntemin benimsenme olasılığını artırır. Kısacası çalışma, *"SNN'leri daha derin ve güçlü yapabiliriz, üstelik enerji avantajıyla birlikte"* mesajını güçlü biçimde vermektedir.
- **Geleceğe Etki ve Yol Açma:** Bu çalışma, SNN'lerin eğitiminde yeni bir paradigmanın kapısını aralamıştır. Gradyan kaybolması sorununun giderilmesiyle, artık araştırmacılar **daha cesur mimariler ve uygulamalar** deneyebilecektir. Örneğin, **görme dışındaki alanlarda** (doğal dil

işleme gibi) spiking ağların derin versiyonları denenebilir, çünkü bilinir ki derinlikten gelecek kazançlar SBP sayesinde elde tutulabilecektir. Ayrıca, bu yöntem diğer tekniklerle (normalizasyon, dönüşüm, vs.) birlikte kullanılarak daha da yüksek performanslar hedeflenebilir. Bu anlamda makalenin yenilikleri, **bir platform teknoloji** gibi düşünülebilir: Başka yeniliklerle birleştirilip SNN'lerin sınırlarını zorlamaya imkan tanır.

Sonuç itibarıyla, “Take A Shortcut Back” makalesi, **SNN'lerin eğitimi konusunda hem kavramsal bir içgörü hem pratik bir çözüm sunmasıyla** önemli ve yenilikçi bir çalışmadır. SNN'lerin önündeki bir bariyeri kaldırarak, bu alandaki ilerlemeyi hızlandırabilecek bir katkı yapmıştır.

## Uygulama Alanları (Applications)

Makalenin ortaya koyduğu yöntem, SNN'lerin eğitim performansını artırdığı için **SNN tabanlı tüm uygulamalarda dolaylı olarak etki gösterebilir**. SNN'lerin halihazırda popüler olduğu veya potansiyel taşıdığı uygulama alanlarına değinecek olursak:

- **Görüntü ve Video İşleme:** Nesne tanıma, nesne takibi, görüntü sınıflandırma, algılama (dedeksiyon) gibi bilgisayarlı görü görevlerinde SNN'ler zaten enerji verimlilikleriyle dikkat çekiyordu <sup>5</sup>. Örneğin, bir **güvenlik kamerası sistemi** SNN kullandığında, hareket olmadığında hesaplama yapmayarak pil ömrünü uzatabilir. SBP yöntemi sayesinde, bu gibi sistemlerde **daha derin spiking ağlar kullanılarak daha yüksek doğruluk** elde edilebilir. Bu da pratikte, SNN'lerin **ANN'lerle yarışacak seviyede** doğruluk sunmasını, böylece düşük güçlü ortamlar için tercih edilebilirliğini artırır. Örneğin, **akıllı telefonlarda veya IoT cihazlarında görüntü tanıma** yaparken SNN kullanmak, SBP yöntemiyle artık daha mantıklı hale gelebilir çünkü doğruluk kaybı yaşamadan enerji tasarrufu sağlanabilir.
- **Otonom Sistemler ve Robotik:** Otonom araçlar veya robotlar, sürekli çevrelerini algılayıp karar vermek zorundadır. Bu sistemlerde enerji verimliliği ve **gerçek zamanlılık** kritiktir. SNN'ler, DVS kameralar gibi **olay tabanlı sensörlerle** birlikte kullanıldığında çok hızlı tepki süreleriyle çalışabilir. Örneğin bir drone'un engellerden kaçınması veya bir endüstriyel robotun anlık hataları algılaması gibi senaryolarda SNN'lerin kullanımı araştırılmaktadır. Bu makalenin yöntemiyle, böyle sistemlerde kullanılan spiking ağlar **daha kompleks (derin) hale getirilerek** karar verme doğruluğu artırılabilir, ancak yine de **gereksiz hesaplamalar yapılmayarak** güç tüketimi sınırlandırılabilir.
- **Nöromorfik Donanımlar:** Intel Loihi veya benzeri nöromorfik çipler, SNN'leri donanım düzeyinde çalıştırmak için tasarlanmıştır <sup>3</sup>. Bu çipler, öğrenmeyi de destekleyebiliyor. SBP yöntemi, teorik olarak bu çiplere de uygulanabilir (eğer eğitim on-chip yapılıyorsa). Ara katman çıktılarından kayıp hesaplamak, donanımda paralel olarak yapılabileceğinden uygun görünüyor. Bu sayede **nöromorfik yongalar üzerinde hızlı ve derin SNN eğitimi** mümkün olabilir. Uygulama olarak, sınırda (edge) çalışan cihazlar kendi kendine öğrenen SNN modelleri kullanabilir. Örneğin, bir akıllı sensör cihazı, ortama adaptasyon için SNN modelini yerinde eğitirken SBP yöntemini kullanarak **daha kısa sürede ve daha başarılı bir öğrenme** gerçekleştirebilir.
- **Zaman Serisi ve İşaret İşleme:** SNN'ler zamansal dinamikleri doğal olarak modelleyebildiği için ses işleme, EEG/EMG gibi biyomedikal işaretlerin analizi, finansal zaman serisi tahmini gibi alanlarda da potansiyel taşırlar. Bu alanlarda yüksek doğruluk için derinlik gerekebilir, ancak eğitim zorluğu SNN kullanımını kısıtlıyordu. Önerilen yöntem, **daha derin ve hassas SNN modellerini bu zaman serisi analizlerinde** kullanılabilir kılarak, örneğin **düşük güçlü giyilebilir cihazlarda gerçek zamanlı sinyal izleme** gibi uygulamalara imkan verebilir.

- **Yapay Zeka Donanımında Enerji Verimli AI:** Genel bir perspektiften, SBP yöntemi, enerji kısıtlı ortamlarda çalışan yapay zeka sistemlerinin yeteneklerini artırır. Örneğin, pille çalışan bir cihazda **her an çevreyi izleyen bir SNN modülü** hayal edelim (sesli komut dinleme, görüntü izleme vb.). Normalde bu modül, doğru çalışması için ya çok basit (yüzeysel) tutulur ya da daha derin ama enerji canavarı bir model kullanılır. SBP sayesinde, **hem derin hem verimli** bir model eğitip cihaza yerleştirmek mümkün olabilir. Bu da akıllı ev cihazlarından tutun da sağlık izleme sensörlerine kadar pek çok cihazın **daha “akıllı” hale gelmesini** sağlar.

Özetle, makalenin buluşları, SNN'lerin kullanıldığı ya da kullanılması öngörülen hemen her alanda dolaylı bir etki yapacaktır. Çünkü bu alanların birçoğunda bugüne dek SNN tercih edilmiyorsa bunun bir sebebi, **yüksek doğruluk elde etmenin zorluğu** idi. Artık bu engel büyük ölçüde aşıldığı için, SNN'ler **düşük güç tüketimi gerektiren uygulamalarda** daha fazla boy gösterebilir. Nesne tanımadan otonom robotlara, biyo-sinyal işlemeden akıllı sensörlere kadar geniş bir yelpazede, bu çalışmanın katkıları sayesinde **daha güçlü spiking sinir ağları görmek mümkün olacaktır**.

## Gelecekte Yapılabilecek Çalışmalar (Future Work)

Makalenin sunduğu yöntem ve bulgular, gelecekte birçok araştırma için zemin hazırlamaktadır. Yazarlar makalede açık bir gelecek işi listesi vermemiş olsalar da, çalışmadan yola çıkarak **ileride yapılabilecek çalışmalar için bazı olası fikirler** şöyle sıralanabilir:

- **Daha Derin ve Farklı Mimari SNN'ler:** Bu çalışmayla, derin SNN'lerin eğitimi önündeki büyük bir engel kalkmış oldu. Gelecekte, **çok daha derin spiking ağlar** (örneğin 100 katmanlı SNN'ler) tasarlanıp SBP yöntemiyle eğitilebilir. Ayrıca, **farklı mimariler** de denenebilir: Örneğin, CNN tabanlı ResNet'ler yerine **spiking Transformer** veya **spiking RNN/LSTM** gibi modellerde gradyan kaybolması sorunu yaşıyorsa, SBP benzeri yaklaşımlarla bunlar eğitilebilir. Bu, SNN'lerin sadece görsel alanla sınırlı kalmayıp dil, kontrol veya diğer sekans verisi işleme alanlarına genişlemesini sağlayabilir.
- **İkame Gradyan Geliştirmeleriyle Birleştirme:** SBP yöntemi gradyan kaybolmasını çözüyor, ancak **gradyan uyumsuzluğu** (mismatch) konusu hala geçerli olabilir. Gelecekte, **daha iyi ikame (surrogate) türev fonksiyonları tasarlayarak** SBP ile birlikte kullanmak, eğitim performansını daha da artırabilir. Örneğin, SBP ile sığ katmanlar sorununu çözerken, aynı zamanda adaptif bir ikame türev yöntemiyle gradyanların doğruluğunu artırmak sinerjik bir etki yapabilir. Bu, belki **öğrenilebilir ikame fonksiyonları** (örn. küçük bir ağ tarafından tahmin edilen türevler) gibi yenilikçi fikirlerle birleşebilir.
- **Denge Katsayısının Optimizasyonu:** Evrimsel eğitimde kullanılan **denge katsayısı  $\lambda$**  için bu çalışmada elle belirlenmiş bir strateji kullanıldı (başlangıç değeri ve azalma şekli sabit). İleride, **bu katsayının daha akıllı ayarlanması** araştırılabilir. Örneğin, **öğrenme sırasında adaptif olarak  $\lambda$ 'ı ayarlayan** bir mekanizma (belki ağın ilerlemesine veya doğrulama performansına bakarak) geliştirilebilir. Hatta  $\lambda$  katsayısını sabit bir formül yerine bir **öğrenme algoritması** (ör. reinforcement learning) ile yönetmek ilginç olabilir. Bu sayede her farklı problem için en uygun denge dinamiği keşfedilebilir.
- **Donanım ve Gerçek Zamanlı Uygulamalar:** Yöntem konsept olarak yazılım seviyesinde sunulmuş olsa da, bunun **donanım uygulamaları** çalışılabilir. Örneğin, nöromorfik çiplerde SBP benzeri bir eğitim desteği nasıl sağlanabilir? Kestirme dalların donanım düzeyinde implementasyonu, bellek ve iletişim açısından optimize edilebilir mi? Bu tip çalışmalar, yöntemin gerçek cihazlarda yaygınlaşması için önemlidir. Ayrıca, gerçek zamanlı sistemlerde (örneğin

otonom araçlarda) SBP kullanarak eğitim yapmadan, **eğitilmiş modeli gerçek zamanda güncelleme** (online learning) gibi konular da incelenebilir.

- **Kuramsal Analizler:** Yöntemin pratikte başarılı olduğu gösterildi, ancak daha derin bir kuramsal analiz de yapılabilir. Örneğin, SBP yöntemi için **konverjans garantileri veya hızları** incelenebilir. Gradyan kaybolması tamamen önlenirse bile, gradient mismatch yüzünden optimizasyon yüzeyinde başka zorluklar çıkıyor mu? Kestirme dalların optimizasyona etkisini matematiksel olarak modellemek, belki *loss landscape* (kayıp yüzeyi) üzerindeki etkilerini analiz etmek, ileride yapılabilecek teorik çalışmalardandır.
- **Yeni Uygulama Alanları Testi:** Gelecekte, yöntemin **farklı uygulamalardaki** etkisini görmek üzere deneyler genişletilebilir. Örneğin, **daha karmaşık veri kümeleri** (video verisi, 3D verisi) üzerinde SBP'li SNN'ler denenebilir. Veya **kontrol ve karar verme** alanında (SNN'lerin kullanıldığı reinforcement learning senaryoları gibi) gradyan akışının iyileştirilmesi ne sağlar, bu incelenebilir. Bu çalışma, büyük ölçüde sınıflandırma problemlerine odaklanmıştı; gelecekte **üretici modeller, tahminci modeller** gibi farklı görevlerde de SBP stratejisinin kullanımı araştırılabilir.
- **Kaybolan Gradyan Dışındaki Sorunlar:** SNN eğitiminde gradyan kaybolması önemliydi ama tek sorun değil. Örneğin, **gradyan patlaması** (exploding gradients) veya **aşırı yüksek ateşleme oranları** gibi konular da zorlayıcı olabilir. SBP, kaybolmayı çözerken patlamayı tetiklemiyor mu, bu dengeler incelenebilir. Belki de gelecekte, **SBP'nin patlayan gradyanlar için de bir tampon görevi** görüp görmediği araştırılabilir ya da ekstra önlemlerle birleştirilebilir.

Özetle, “Take A Shortcut Back” makalesinin açtığı yol oldukça geniştir. Gerek SNN'lerin kendini geliştirmesi, gerek farklı yöntemlerle kombinasyon, gerekse kuramsal ve uygulamalı yeni araştırmalar açısından pek çok olasılık bulunmaktadır. Bu fikirlerin hayata geçirilmesiyle, SNN'lerin yapay zeka içindeki rolü daha da pekişecek ve **yüksek verimli, güçlü sinir ağları** vizyonuna bir adım daha yaklaşılacaktır.

---

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29  
30 31 32 36 37 38 39 40 48 49 99 100 104 arxiv.org

<https://arxiv.org/pdf/2401.04486>

33 34 35 41 42 43 44 45 46 47 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68  
69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97

98 101 102 103 [2401.04486] Take A Shortcut Back: Mitigating the Gradient Vanishing for Training Spiking Neural Networks

<https://arxiv.org/pdf/2401.04486>